CONTINUOUS PHASE MODULATION FOR HIGH SPEED FIBER-OPTIC LINKS

A Thesis
Presented to
The Academic Faculty

by

Thomas Frederick Detwiler

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2011

CONTINUOUS PHASE MODULATION FOR

HIGH SPEED FIBER-OPTIC LINKS

Approved by:

Dr. Stephen E. Ralph, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Gee Kung Chang
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. John Barry, Co-Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Robert Lingle, Jr.
OFS and School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Gordon Stüber
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Michael Chapman
School of Physics
Georgia Institute of Technology

Date Approved: November 8, 2011

To my wife Alicia and my mother Elaine.

ACKNOWLEDGEMENTS

Many thanks go to a great number of people, without whom this work would not have been possible. The sacrifices made on my behalf are far greater than I will ever repay. The first thanks go to my bride Alicia, who makes everything possible, keeps me driving toward my goals, and has given so much to ensure all our family's goals are met. Thanks to Dr. Andreas Cangellaris and Dr. Douglas Jones of University of Illinois for urging to pursue a PhD. Also, I recognize Adtran for their encouragement and support throughout the entire degree process, and exceptional treatment of all employees in the Adtran family. I thank Dr. Stephen Ralph and Dr. John Barry for their patience, guidance, and willingness to work with me throughout the degree process, given the uncertainty and remote collaboration necessary to fulfill the degree requirements. Thanks to Bert Basch for his guidance and financial support by Verizon while I worked as a GRA. Bert's suggestions supplied a great amount of material for this dissertation. And finally I thank Dr. Gordon Stüber, Dr. Gee-Kung Chang, Dr. Michael Chapman, and Dr. Robert Lingle, Jr. of OFS for serving on my reading committee to ensure the quality of this work.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Fiber-optic networks are continually evolving to accommodate the ever increasing data rates demanded by modern applications and devices. The current state-of-the art systems are being deployed with 100 Gb/s rates per wavelength while maintaining the 50 GHz channel spacing established for 10 Gb/s dense wavelength division multiplexed (DWDM) systems. Phase modulation formats (in particular quadrature phase shift keying – QPSK) are necessary to meet the spectral efficiency (SE) requirements of the application. The main challenge for phase modulated optical systems is fiber nonlinearities, where changes in intensity of the combined optical signal result in changes to the fiber's refractive index. Limiting launch power is the primary means to avoid dramatic intensity fluctuations, a strategy which in turn limits the available signal-to-noise ratio (SNR) within the channel.

Continuous phase modulation (CPM) is a format in which data is encoded in the phase, while the amplitude is constant throughout all transmission (even during transitions between symbols). With the goal of reducing the impact of nonlinearities, the purpose of this research was to identify a set of CPM signals best suited for high speed fiber-optic transmission, and quantify their performance against other formats. The secondary goal was to identify techniques appropriate for demodulation of high speed fiber-optic systems and implement them for simulation and experimental research.

CPM encompasses a number of variable parameters that combine to form an infinite number of unique schemes, each of which is characterized by its own SE, minimum distance, and implementation complexity. A method for computing minimum

distance of DWDM-filtered CPM formats is presented and utilized to narrow down to a range of candidate schemes. A novel transmitter design is presented for CPM signal generation, as well as a number of novel reception techniques to achieve proper demodulation of the CPM signal from the coherent optical receiver. Using these methods, the identified range of candidate schemes was compared in simulation to the conventional QPSK format, showing that some modest gain can be expected from CPM.

Through these and other simulations, it is revealed that fiber nonlinearities depend on the aggregate sum of all wavelengths rather than the imposition of each separate carrier on its neighbors. Therefore the constant envelope of CPM does not directly impact the nonlinearities since multiple carriers will photonically interfere and result in intensity fluctuations regardless of modulation format. Additionally, dispersive effects in fiber decompose the underlying channels so that the intensity throughout propagation is nearly Gaussian distributed, regardless of format. The benefits gained from CPM are thus limited to schemes that attain a higher minimum distance than alternative formats (in the given channel passband), and for optically compensated links in which low dispersion is maintained throughout the fiber link.

CHAPTER 1

INTRODUCTION

Fiber-optic communication systems are currently evolving to incorporate signal processing and phase modulation techniques that enable higher spectral efficiency (SE) and performance than binary intensity modulation with direct detection (IMDD). This trend comprises digital signal processing techniques at the transmitter and receiver, advancements in laser linewidth, fiber manufacturing, optical filtering, error correction, and the fulfillment of 1990's era research in coherent optical reception. With these techniques, network operators will attempt to keep pace with demand for data throughput by scaling existing 10 Gb/s or 40 Gb/s links to 100 Gb/s in the near future, and 400 Gb/s and 1 Tb/s within the next decade.

A number of challenges must be overcome to achieve these data rates. One fundamental issue is nonlinear refraction, whereby the phase of the signals propagating through fiber is modulated proportionally to intensity. This effect limits the launch power, thereby limiting the available optical signal-to-noise ratio (OSNR) at the receiver, making it the fundamental capacity limiting effect. Another challenge is the extreme nature of the bit and symbol rates, which requires special consideration for the underlying electronic and optical signal processing architectures. While overcoming these challenges requires several breakthroughs, it is nevertheless important to emphasize that the eventual adoption of these techniques depends upon market economics, i.e. meeting throughput and latency demands at a minimal cost per bit.

A main focus of this research is to study the mitigation of fiber nonlinear effects by means of choice of modulation format. Since nonlinear phase modulation is proportional to intensity, one strategy would be to minimize amplitude fluctuations of the optical carrier. Continuous phase modulation (CPM) is a modulation format which maintains constant amplitude throughout all transmission, and encompasses a broad range of parameters that can be used to uniquely determine SE and performance. By exploiting the constant amplitude feature in the nonlinear fiber media, it is possible to enable higher launch power than alternative formats. Making use of this feature, CPM is capable of achieving higher channel capacity than alternative formats (e.g. quadrature phase shift keying – QPSK) in fiber types with high nonlinear coefficients.

1.1 Organization and Scope

The purpose of this dissertation is two-fold. The main objective is to lay the foundation for exploring alternative modulation formats for fiber-optics, with particular attention to their behavior as total launch power increases nonlinear effects dominate performance. Secondly, the manuscript is intended to serve as a reference for future developers of the 100G Consortium's modulation and demodulation code, which is central to gaining insight to modulation formats. It is noted that CPM typically lies outside the range of expertise of fiber-optic communications experts, and conversely that methods for modulation and coherent detection in the optical domain are not commonly known by communication DSP experts. Therefore, the narrative throughout the dissertation intended to provide an in-depth review of the fundamentals of fiber-optics, CPM, and demodulation DSP techniques, in addition to elaborating on the exploration of CPM and alternative formats in a high speed fiber-optic application.

This dissertation is arranged in six subsequent chapters and three appendices. Chapter 2 provides the necessary background information for fiber communication systems, including the channel description and an overview of underlying components and methodologies for modulating and receiving an optical carrier. An overview of contemporary research trends is included, as is a summary of the recent efforts of the Georgia Tech 100G Consortium.

In Ch. 3, the CPM format is reviewed with a thorough examination of the defining signal parameters and the characteristics that result from particular choices of parameters (e.g. SE and minimum distance). This is followed by a general introduction to CPM modulation and demodulation techniques and a review of recent CPM research.

Chapter 4 contains a detailed description of each of the techniques used in the 100G Consortium's demodulation code. This includes six separate stages; chromatic dispersion compensation, polarization mode demultiplexing, symbol timing recovery, channel equalization, carrier phase recovery, and pattern matching. Since each module typically has a variety of applicable techniques, particular attention is given to justify the choice of each method.

In Ch. 5 the criteria for selecting CPM format is explored in depth, narrowing the vast parameter space to specify a small range of modulation formats of interest for high speed fiber-optic communications. The techniques specific to CPM signal generation and reception over fiber are then revealed in depth, which enable performance to be quantified via simulation.

In Ch. 6, the focus turns to simulation and experimental efforts for exploring CPM, QPSK, and offset QPSK. Beginning with CPM simulations, candidate schemes

are tested in a variety of link configurations and fiber types, giving insight to scenarios in which CPM may be beneficial in fiber. A number of QPSK experiments are described to exhibit the use of the demodulation code in the 100G Consortium lab environment. Offset QPSK experimental results are provided, showing the effects of an alternative modulation scheme that is achievable with existing QPSK components. Finally, an analysis of QPSK formats (including a constant intensity variation) is performed in simulation for a 1 Tb/s system in the zero-dispersion 1310 nm band. This provides further insight and leads to deeper discussions of the mechanisms behind nonlinear interactions in a multi-channel fiber environment.

The dissertation concludes with a summary of the findings in Ch. 7, and points to new areas of interest prompted by these efforts. Three appendices follow the conclusion, containing the 100G Consortium demodulation code for polarization multiplexed QPSK, a in-depth derivation of the block mode CPM receiver, and the code for CPM signal generation and reception.

CHAPTER 2

FIBER COMMUNICATION SYSTEMS

2.1 Evolution of Fiber Communications

Throughout the first thirty years of optical fiber telecommunications, deployed systems utilized intensity modulation to achieve data rates from 10's of Mb/s to 10's of Gb/s per optical carrier [1]. In the mid 1970's, the earliest fiber-optic communication systems were deployed over multimode fiber with a single carrier in the 800 nm band, with electro-optical repeaters spaced at 10 km intervals at bit rates from 32-140 Mb/s [2]. By the mid-1980's, single-mode fiber (SMF) was deployed for systems in the low-dispersion 1310 nm band. In the late 1980's, erbium-doped fiber amplifiers (EDFAs) provided a breakthrough in optical signal regeneration in the low-loss 1550 nm band. The higher dispersion in the 1550 nm band, which was originally perceived as the primary performance limitation, was compensated by the use of dispersion shifted fibers (DSF), non-zero dispersion shifted fiber (NZDF), and dispersion compensating fibers (DCF). In the 1550 nm band dispersion of DSF is close to zero, which was originally desired avoid dispersion and extend reach. However excessive nonlinear effects arose from the use of DSF prompting the use of NZDF with roughly 4.5 ps/nm-km dispersion. DCF is used to optically compensate for CD with roughly -80 ps/nm-km dispersion. Though each of these fiber types has lower (or negative) dispersion compared to standard

SMF (SSMF) (17 ps/nm-km) each also has a lower effective area, which corresponds to higher nonlinear coefficient.

EDFAs underwent massive deployment in the early 1990's, providing the framework for signal amplification across a very large band, and with this wavelength-division multiplexing (WDM) emerged. WDM systems feature several lasers transmitting at different wavelengths, all over the same fiber medium. In the 2000's, advancements in high-speed digital circuitry and forward error correction (FEC) were applied to WDM systems to achieve per-wavelength rates up to 40 Gb/s.

Current commercial WDM fiber communication systems are capable of achieving bit rates in excess of 3 Tb/s [3], and higher rates have been demonstrated experimentally [4]. To date, advancements in fiber-optic technology have fueled explosive growth in network bandwidth demand. In order to maintain this trend, research is now focused on increasing capacity through phase and quadrature modulation, coherent reception, and polarization multiplexing (POLMUX) [4-7]. Upcoming standards [8] for 100 Gb/s (payload rate) transmission per wavelength are based on these efforts, and intended for operation in conjunction with existing infrastructure, including the desire to maintain reach of installed 10 Gb/s systems.

Optical network standards have historically been divided into two applications, namely telecom (SONET and SDH) and datacom (Ethernet). SONET and SDH evolved from T-carrier and E-carrier circuit switched telecom standards [9], and were both based on a synchronous network in multiples of 155.52 Mb/s. Ethernet standards have evolved exponentially in decades of 10 Mb/s, are packet-based, connectionless, and incompatible with SONET/SDH above the physical layer (i.e. framing formats and data link layers

differ [9]). Fortunately, these two paths merged very close to 10 Gb/s in the early 2000's, providing an economy of scale for equipment capable of either standard. Consequentially, most deployed equipment operates at 10 Gb/s. However, techniques for improving OOK modulation slowed. Though 40 Gb/s is available it may be deployed as OOK or BPSK and does not enjoy as widespread use as 10 Gb/s. The target research platform for future telecom and datacom transmission is now 100 Gb/s, with attention currently turning towards 400 Gb/s and 1 Tb/s [10-12].

2.2 Fiber Communication Channel

Optical fiber is a waveguide for electromagnetic propagation with low loss for wavelengths between approximately 900 nm and 1700 nm. Fiber is an excellent medium for point-to-point communications due to its bandwidth, loss, flexibility, size, security, immunity to electromagnetic interference, and the abundance of sand for its manufacture. Several effects characterize the communication channel of optical fiber, most of which are assumed to be nearly time-invariant. The effects of primary concern for this study are attenuation, chromatic dispersion, polarization mode dispersion, and nonlinear refraction in SMF.

Attenuation in fiber is very low as compared to propagation of electromagnetic waves through other materials [13]. When light is launched into a fiber at a sufficiently low angle of incidence it propagates via total internal reflection, and very little signal power is lost to the material. Loss in a standard SMF link increases proportionally to distance, and is wavelength-dependent with relative minima near 1310 nm and 1550 nm. A peak near 1380 nm is due to hydroxyl ion (OH^-) impurities in the fiber due to water vapor, though modern manufacturing techniques (those used in AllWave® fiber) nearly

eliminate this effect [14]. It is appropriate to assume flat, time-invariant attenuation over each channel in WDM configurations with channel spacings of 50 to 100 GHz. Note that the loss is typically 0.32 dB/km at 1310 nm and is typically 0.19 dB/km at 1550 nm in the AllWave® ZWP Flex SMF in use in the Georgia Tech 100G Consortium lab.

2.2.1 Dispersion and the nonlinear Schrödinger equation

Pulse propagation in fiber is described by the wave equation, and when elaborated to include nonlinear and dispersive terms that exist in fiber, is known as the nonlinear Schrödinger equation (NLS) [14]. Consider a pulse with electric field

$$\varepsilon(z,t) = \frac{1}{2} E_0(z,t) \exp(\ j\omega_0 t - j\beta(\omega) z) + c.c., \tag{2.1}$$

where the *phase constant* of the signal $\beta(\omega)=n_0(\omega)/k_0$ which varies nonlinearly ($n_0(\omega)$ is the refractive index, and k_0 is the wave number). The Taylor series expansion of the phase constant about the optical frequency ω_0 is:

$$\beta(\omega) = n_0(\omega)k_0 = \beta_0 + \beta_1(\omega - \omega_0) + \frac{1}{2}\beta_2(\omega - \omega_0)^2 + \frac{1}{6}\beta_3(\omega - \omega_0)^3 + \ldots \quad \text{and} \quad \beta_n = \frac{\partial^n \beta(\omega)}{\partial \omega^n}. \tag{2.2}$$

Since the pulse is time dependent and the velocity of each spectral component is a function of frequency, the pulse can be expressed as the Fourier transform of its spectrum as:

$$\varepsilon(z,t) = \int_{-\infty}^{\infty} A(\omega - \omega_0) \exp(\ j\omega_0 t - j\beta(\omega) z) d\omega + c.c., \text{ where} \tag{2.3}$$

$$E_0(z,t) = \int_{-\infty}^{\infty} A(\omega - \omega_0) \exp\{\ j(\omega - \omega_0)t\} \exp\{\ -j(\beta - \beta_0)z\} d(\omega - \omega_0) + c.c. \tag{2.4}$$

Substituting the Taylor expansion (2.2) into (2.4) gives

$$E_0(z,t) = \int_{-\infty}^{\infty} A(\omega-\omega_0)\exp\{j(\omega-\omega_0)t\}\exp\{-j(\omega-\omega_0)\beta_1 z\}\exp\left\{-\frac{j}{2}(\omega-\omega_0)^2\beta_2 z\right\} \times \exp\left\{-\frac{j}{6}(\omega-\omega_0)^3\beta_3 z\right\} d(\omega-\omega_0) + c.c. \tag{2.5}$$

Ordinarily one would simply substitute (2.5) into the wave equation. Here the objective is to identify the separate terms for propagation velocity and dispersion, so the next step is to first take the time derivative, then insert the result into the spatial derivative [14]:

$$\frac{\partial E_0(z,t)}{\partial t} = \int_{-\infty}^{\infty} j(\omega-\omega_0)A(\omega-\omega_0)\exp\{j(\omega-\omega_0)t\}\exp\{-j(\omega-\omega_0)\beta_1 z\} \times \exp\left\{-\frac{j}{2}(\omega-\omega_0)^2\beta_2 z\right\}\exp\left\{-\frac{j}{6}(\omega-\omega_0)^3\beta_3 z\right\} d(\omega-\omega_0) + c.c. \tag{2.6}$$

$$\frac{\partial E_0(z,t)}{\partial z} = \int_{-\infty}^{\infty} -j\beta_1(\omega-\omega_0)A(\omega-\omega_0)\exp\{j(\omega-\omega_0)t\}\exp\{-j(\omega-\omega_0)\beta_1 z\} \times \exp\left\{-\frac{j}{2}(\omega-\omega_0)^2\beta_2 z\right\}\exp\left\{-\frac{j}{6}(\omega-\omega_0)^3\beta_3 z\right\} d(\omega-\omega_0) + c.c. \tag{2.7}$$

$$+ \int_{-\infty}^{\infty} -j\frac{\beta_2}{2}(\omega-\omega_0)^2 A(\omega-\omega_0)\exp\{j(\omega-\omega_0)t\}\exp\{-j(\omega-\omega_0)\beta_1 z\} \times \exp\left\{-\frac{j}{2}(\omega-\omega_0)^2\beta_2 z\right\}\exp\left\{-\frac{j}{6}(\omega-\omega_0)^3\beta_3 z\right\} d(\omega-\omega_0) + c.c.$$

$$+ \int_{-\infty}^{\infty} -j\frac{\beta_3}{6}(\omega-\omega_0)^3 A(\omega-\omega_0)\exp\{j(\omega-\omega_0)t\}\exp\{-j(\omega-\omega_0)\beta_1 z\} \times \exp\left\{-\frac{j}{2}(\omega-\omega_0)^2\beta_2 z\right\}\exp\left\{-\frac{j}{6}(\omega-\omega_0)^3\beta_3 z\right\} d(\omega-\omega_0) + c.c. \quad ,$$

which altogether gives the result

$$\frac{\partial E_0(z,t)}{\partial z} + \beta_1\frac{\partial E_0(z,t)}{\partial t} = j\frac{\beta_2}{2}\frac{\partial^2 E_0(z,t)}{\partial t^2} + \frac{\beta_3}{6}\frac{\partial^3 E_0(z,t)}{\partial t^3} \tag{2.8}$$

Though (2.8) is an intermediate step to arrive at the NLS, it completely describes dispersion in the fiber, which is the main linear effect. The term β_1 determines the *group-velocity dispersion* (GVD) $v_g = \beta_1^{-1}$. The frequency dependence of the GVD term causes

pulses propagating through fiber to spread in time. Accordingly, the *dispersion parameter D* relates to β_2 as follows [15]:

$$D = \frac{d}{d\lambda} v_g^{-1} = \frac{-2\pi c}{\lambda^2} \beta_2 \tag{2.9}$$

Note that *D* is specified for any fiber type, and is expressed in units of ps/nm-km. The β_3 term is also specified as *dispersion slope* for the effect it has on broad channels (i.e. the rate of change of *D*) or adjacent channels.

The term chromatic dispersion (CD) encompasses material, waveguide, and profile dispersion (listed in order of decreasing impact). Material dispersion arises from frequency dependence of the refractive index in the silica material. Waveguide dispersion relates to the power distribution of the propagation mode (between the core and the cladding), which is also frequency dependent. Profile dispersion is caused by frequency dependence between the *relative index difference* between the core and the cladding. For the communication channel, CD can be modeled to first order as the all-pass transfer function [16]:

$$H_D(\omega) = \exp\left\{ j \frac{\lambda_0^2 DL\ \omega^2}{4\pi c} \right\} \tag{2.10}$$

CD varies across the spectra in standard SMF, as shown in Fig. 2.1 [14]. Interestingly, CD is nearly zero in the 1310 nm band, and is significantly higher (17 ps/nm-km) in the 1550 nm band. Besides standard SMF, DSF is also available to achieve lower or negative dispersion in the 1550 nm region. TrueWave® RS LWP Fiber (one such NZDF in the 100G lab) has CD on the order of 4 ps/nm-km in the 1550 nm band. Compensation of CD is an essential channel equalization requirement for links spanning appreciable length.

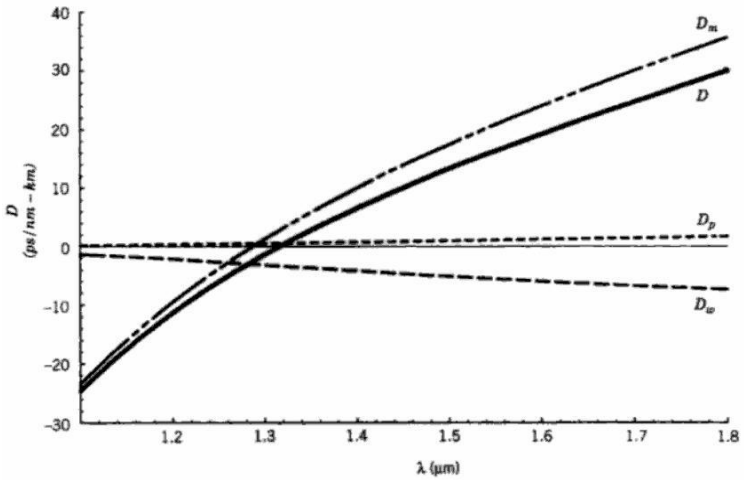


Fig. 2.1. Dispersion versus wavelength for SSMF [14, pp. 152], illustrating the impact of the separate groupings of composite material (D_m), waveguide (D_w), and profile (D_p) dispersion

Nonlinear refraction is another significant characteristic of the optical fiber channel. The index of refraction in fiber is dependent on the intensity of the optical signals being carried. Since the speed of propagation is proportional to the index of refraction, this effect causes the phase of the signal at the receiver to vary depending on the intensity of the transmitted signal. In a WDM environment, each channel's intensity fluctuation causes phase modulation within its own band (called self phase modulation – SPM) as well as those with nearby wavelengths (called cross phase modulation – XPM). Note that in SPM and XPM no energy transfer takes place, only signal perturbation. The derivation of the equations for nonlinear refraction is based on the macroscopic medium polarization vector and its power series expansion of the electric field vector, and is available in [14, pp. 241-242]. The resulting equation for the intensity-dependent refractive index of SPM is:

$$n = n_0 + n_2' \left| E_0(z,t) \right|^2 = n_0 + \left(\frac{2\eta_0}{n_0} \right) n_2' I(z,t) \tag{2.11}$$

The parameter n_2' is called the *nonlinear refractive index* and is proportional to the third-order susceptibility tensor of the fiber media. The wave equation applied to a nonlinear media with time varying index is [14]:

$$\frac{\partial^2 \varepsilon(z,t)}{\partial z^2} - \frac{\partial^2}{\partial t^2}\left[\mu_0 \in_0 n^2 \varepsilon(z,t)\right] = 0 \tag{2.12}$$

Assuming n_2' is small, the n^2 may be approximated as $n^2 \approx n_0^2 + 2n_0 n_2' \left|E_0(z,t)\right|^2$, which substitutes into (2.12) to yield

$$\frac{\partial^2 \varepsilon(z,t)}{\partial z^2} - \mu_0 \in_0 n_0^2 \frac{\partial^2 \varepsilon(z,t)}{\partial t^2} = 2\mu_0 \in_0 n_0 n_2' \frac{\partial^2}{\partial t^2}\left[\left|E_0(z,t)\right|^2 \left|\varepsilon(z,t)\right|\right]. \tag{2.13}$$

Given that $\beta_1 = n_0\sqrt{\mu_0 \in_0}$, (2.1) is substituted into (2.13) and the slowly varying envelope approximation results in a first-order equation to incorporate the nonlinearity:

$$\frac{\partial E_0(z,t)}{\partial z} + \beta_1 \frac{\partial E_0(z,t)}{\partial t} = -j\frac{n_2'}{n_0}\beta_0 \left|E_0(z,t)\right|^2 E_0(z,t). \tag{2.14}$$

The amount of SPM-induced phase shift of a propagating electric field is then given as

$$\delta\phi(z,t) = n_2' k_0 \left|E_0(z,t)\right|^2 z. \tag{2.15}$$

The impact of this phase distortion, though less significant for intensity modulated schemes (in which data is encoded in amplitude, not phase), has the potential of dramatically altering the state of phase modulated signals.

Finally, under the previous assumption that n_2' and β_2 are small, the right-hand sides of (2.8) and (2.14) can be added to account for both linear and nonlinear propagation interactions. The NLS includes both effects and a loss term proportional to the loss coefficient α. By convention, the frame of reference follows the pulse as it traverses the fiber ($t_L = t - \beta_1 z$) to yield the final form of the NLS equation:

$$\frac{\partial E_0(z,t_L)}{\partial z} = -\frac{\alpha}{2}E_0(z,t_L) + j\frac{\beta_2}{2}\frac{\partial^2 E_0(z,t_L)}{\partial t_L^2} + j\frac{\beta_3}{6}\frac{\partial^3 E_0(z,t_L)}{\partial t_L^3} - j\frac{n_2'}{n_0}\beta_0\left|E_0(z,t_L)\right|^2 E_0(z,t_L) \tag{2.16}$$

The first line of (2.16) corresponds to fiber loss (absorption), the second represents the GVD, and the third is SPM.

Nonlinear refraction in the form of XPM is formulated similarly to (2.11):

$$n = n_0 + 2n_2'\left|E_{0,1}(z,t)\right|^2 = n_0 + \left(\frac{2\eta_0}{n_0}\right)n_2' I_1(z,t) \tag{2.17}$$

Here the subscript '1' indicates that the modulation in the refractive index comes from intensity variation in the adjacent channel. Note that (2.7) assumes co-polarized waves and has twice the impact as SPM, though this is reduced by a factor of 1/3 for the cross-polarized case. The difference in contribution between co-polarized and cross-polarized fields is due to the third-order susceptibility (fourth-rank tensor) of the transparent isotropic fiber media [17]. Briefly, the birefringent nonlinear polarization components P_x and P_y are given by:

$$P_i = \frac{3\epsilon_0}{4}\sum_j\left(\chi^{(3)}_{xxyy}E_iE_jE_j^* + \chi^{(3)}_{xyxy}E_jE_iE_j^* + \chi^{(3)}_{xyyx}E_jE_jE_i^*\right), \text{ where} \tag{2.18}$$

$$\chi^{(3)}_{xxxx} = \chi^{(3)}_{xxyy} + \chi^{(3)}_{xyxy} + \chi^{(3)}_{xyyx} \approx 3\chi^{(3)}_{xxyy} \approx 3\chi^{(3)}_{xyxy} \approx 3\chi^{(3)}_{xyyx}\ . \tag{2.19}$$

The three susceptibility elements of (2.18) are nearly the same magnitude in silica fiber, as indicated in (2.19). If they are identical, then using (2.18) P_x and P_y simplify to:

$$P_x = \frac{3\epsilon_0}{4}\chi^{(3)}_{xxxx}\left[\left(\left|E_x\right|^2 + \frac{2}{3}\left|E_y\right|^2\right)E_x + \frac{1}{3}\left(E_x^*E_y\right)E_y\right] \tag{2.20}$$

$$P_y = \frac{3\epsilon_0}{4}\chi^{(3)}_{xxxx}\left[\left(\left|E_y\right|^2 + \frac{2}{3}\left|E_x\right|^2\right)E_y + \frac{1}{3}\left(E_y^*E_x\right)E_x\right] \tag{2.21}$$

The interaction length between pulses on adjacent channels in fiber is significantly reduced by the difference in group velocity of the channels and dispersion. The interaction length is approximated by the *walk-off length*:

$$L_W = T_0/|\beta_1(\lambda_1) - \beta_1(\lambda_2)|^2, \quad (2.22)$$

where T_0 is the pulse width of interest. Furthermore, fluctuations in intensity decrease as the signals attenuate through a span, so the role of nonlinear refraction is most significant near the launch point. The length over which the signal has significant power is the *effective length*:

$$L_{eff}=[1-\exp(-\alpha L)]/\alpha, \quad (2.23)$$

where α is the loss coefficient of the fiber.

The degree of severity of SPM and XPM is dependent on fiber characteristics; in specific the nonlinear refractive index n_2' is proportional to the ratio γ/A_{eff}, where γ is the fiber's nonlinear parameter and A_{eff} is the effective core area.

It is possible to compensate for SPM by applying an opposing phase rotation as the intensity increases, either at the transmitter (via predistortion) or receiver. However, WDM network configurations that insert or terminate wavelengths at different sites along a link are incapable of determining XPM interactions. Therefore XPM effects are generally treated as a random phase noise process, which restricts the allowed launch power (and therefore limits fiber channel capacity).

Note that XPM calculation is based on the ensemble intensity of all co-propagating channels as they interfere with each other through the fiber. Thus, constant intensity per channel is not sufficient to avoid nonlinear refraction when the number of random phase modulated carriers is greater than two. However, the peak ensemble

intensity is partially dependent upon the peak intensity of each carrier, so minimizing the peak-to-average power ratio can provide some benefit for avoiding nonlinearities.

Another form of dispersion in fiber is polarization mode dispersion (PMD). Electric fields propagating along the linear polarization axes in a fiber would travel at the same rate if it were perfectly cylindrical. Fiber is nearly cylindrical, but the cross section at any point in the fiber is slightly elliptic depending on manufacturing and external stress. Modern fiber manufacturing spins the fiber as it is drawn to average out the elliptic orientation, so PMD is not as significant as CD. The differential group delay (DGD) between the two orthogonal states of polarization at the fiber output is proportional to the square-root length [18]:

$$\tau_{DGD} = D_{PMD}\sqrt{L} \tag{2.24}$$

The parameter D_{PMD} is typically specified by fiber manufacturers for each product, and is on the order of 0.1 $\text{ps}/\sqrt{\text{km}}$. Unlike CD, the DGD is time varying since D_{PMD} is partially dependent on the ambient temperature and mechanical stress of the fiber. It is possible to multiplex and demultiplex two independent modulated optical signals over fiber since the modes propagate orthogonally, albeit at different rates. Propagation of E-fields through fiber is characterized by the Jones matrix:

$$T = \begin{bmatrix} \sqrt{\alpha}\, e^{j\delta} & -\sqrt{1-\alpha} \\ \sqrt{1-\alpha} & \sqrt{\alpha}\, e^{-j\delta} \end{bmatrix}, \tag{2.25}$$

where α is the power splitting ratio and δ is the phase difference between the two modes. The Jones matrix is a transfer function between the input and output E-field vectors, i.e. $\boldsymbol{E}_{\text{out}} = \boldsymbol{T}\cdot\boldsymbol{E}_{\text{in}}$.

2.3 Fiber-Optic System Components

The performance of components of optical communications system impose restrictions on the link, particularly with respect to bandwidth. The components of particular importance to this study are the transmitter laser, external electro-optic modulator, WDM channel filter, EDFA, differential receiver, and coherent optical receiver.

Current high-speed optical transmitters employ a tunable laser source with an external modulator rather than merely modulating the laser directly. For the lowest achievable linewidth, a proxy for phase noise, the choice of an external cavity laser (ECL) is most appropriate. ECL lasers are capable of generating continuous wave signals with linewidths on the order of 10 kHz [19] though 100 kHz linewidths are and anticipated specification for typical coherent systems (for transmit carrier and receiver local oscillator) [5].

External modulation in contemporary transmitters is performed in active electro-optical waveguides. The material of choice for optical modulation is lithium niobate ($LiNbO_3$) [15]. The strength of an electric field across a lithium niobate waveguide determines its index of refraction, therefore electrodes are placed in parallel along both sides of the waveguide channel. Nearly all photons entering the waveguide from the laser are in phase with each other, so fluctuations on the electrode drive voltage result in phase shifts at the output of the waveguide. This configuration, of a single waveguide channel between electrodes, is used as a phase modulator.

In addition to the phase modulator, the other predominant modulator structure is the Mach-Zehnder modulator (MZM). The MZM comprises two phase modulators in

parallel, with electrodes along each waveguide segment as shown in Fig. 2.2a. The electric field transfer function of an MZM is

$$\begin{aligned}
\frac{E_{out}}{E_{in}} &= \frac{1}{2}\left[\exp\left(j\pi\frac{V_1}{V_\pi}\right)+\exp\left(j\pi\frac{V_2}{V_\pi}\right)\right] \\
&= \frac{1}{2}\left[\exp\left(j\pi\frac{V_C+V_D/2}{V_\pi}\right)+\exp\left(j\pi\frac{V_C-V_D/2}{V_\pi}\right)\right] \\
&= \frac{1}{2}\left[\cos\left(\pi\frac{V_C+V_D/2}{V_\pi}\right)+\cos\left(\pi\frac{V_C-V_D/2}{V_\pi}\right)+j\sin\left(\pi\frac{V_C+V_D/2}{V_\pi}\right)+j\sin\left(\pi\frac{V_C-V_D/2}{V_\pi}\right)\right] \\
&= \cos\left(\pi\frac{V_D}{2V_\pi}\right)\left[\cos\left(\pi\frac{V_C}{V_\pi}\right)+j\sin\left(\pi\frac{V_C}{V_\pi}\right)\right]
\end{aligned}, \quad (2.26)$$

where $V_C=(V_1+V_2)/2$ is the common-mode voltage and $V_D=V_1-V_2$ is the differential voltage. One common configuration of an MZM is to drive the control voltage for both arms from the same source in opposition, where $V_1=V_{CTL}=-V_2$ (i.e. $V_C=0$, $V_D=2V_{CTL}$). In such a configuration, the MZM output is either in-phase or 180° out-of-phase with respect to the input, with variable amplitude, i.e. $E_{out} = E_{in}\cdot\cos(\pi V_{CTL}/V_\pi)$. It is important to note the nonlinear (cosine) relationship between the control voltage and the output electric field. Quadrature modulation is achieved by placing two independent MZMs in parallel with a 90° phase shift inserted at either output prior to recombination (see Fig. 2.2b). This must be implemented as an integrated device to assure path matching along the optical waveguide and electronic path, over temperature and operating conditions.

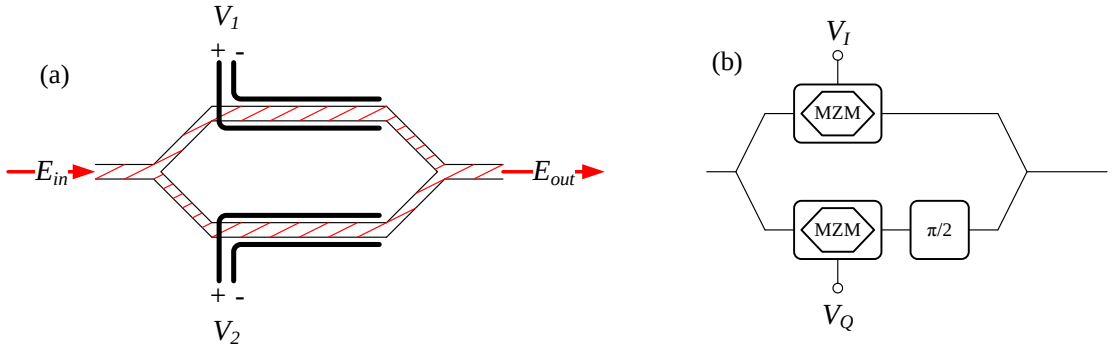


Fig 2.2. Mach-Zehnder modulator structures. Individually (a) and paired in quadrature configuration (b).

Modulated optical signals propagate through WDM networks between endpoints. Typically, optical links pass through reconfigurable optical add/drop multiplexers (ROADMs) prior to optical-to-electrical conversion and termination. The ROADM facilities use optical filters to select channels from a WDM data link to drop from the aggregate received signal, as well as inserting channels to the aggregate transmitted signal. ROADM filters typically have multiple bandwidth settings to operate over channels arranged in 50 GHz increments (also called a *50 GHz grid*) or 100 GHz increments (*100 GHz grid*). Optical channel filters typically feature a super-Gaussian transfer function [20], modeled as:

$$H_{SG}(f) = \exp\left\{-\ln(2)\left\lceil\frac{|f|}{B}\right\rceil^{2N}\right\} \tag{2.27}$$

where N is the order of the filter (typical values of commercial optical filters are 4 to 6), and B is the bandwidth. Note that the transfer function attenuation in (2.27) is not normalized. ROADMs are capable of forwarding channels in addition to adding or dropping, so each WDM channel is subjected to a cascade of ROADM filters. Noting also that variation may exist in the center frequencies of the filter set, the cascaded filters are expected to have sharp rolloff with narrower bandwidth.

The EDFA is another component playing a vital role in WDM communication systems. An EDFA comprises a short length of fiber (on the order of ten meters) doped with the rare earth element erbium. Erbium is used since it provides wideband gain over the wavelengths 1525-1570 nm [15]. The EDFA is pumped by high-power semiconductor lasers at either 980 or 1480 nm to populate the higher energy levels. Incoming photons stimulate atoms in the higher energy state, emitting additional photons

at the same energy (same frequency) and phase and therefore amplify the optical signal without any electro-optic conversion or regeneration. However, stimulated emissions are not the only mechanism for energy release in the 1550 nm band for the atoms. Spontaneous emissions of photons that are incoherent with the incoming signals also occur, as well as the amplification of spontaneous emissions from prior EDFAs. This latter effect, dubbed amplified spontaneous emission (ASE), is recognized as a performance-limiting noise component in optical systems. ASE typically overwhelms the receiver thermal noise contribution in optically amplified links, and so in optical systems the ratio of optical in-band signal power to the ASE noise floor at the receiver input, the optical signal-to-noise ratio (OSNR), is the prevailing figure of merit for link margin. By convention, OSNR is defined as the ratio of total signal power to the two-sided noise measured in 0.1 nm of optical spectrum [21]. Note that ASE may be modeled as a white Gaussian process, but since it occurs in the channel it is colored by the cascade of filters following each EDFA.

After the transmitted signal passes through the WDM network, it can be selected by an optical filter at the terminating site. At this point, the signal may be detected directly, differentially or coherently. A direct detection receiver is a photodetector and amplifier, capable of discerning between amplitude levels of a binary OOK signal. Under differential detection the phase states of adjacent symbols are combined in an interferometer to achieve constructive or destructive interference. In doing so, a phase format is converted at the receiver into one or more amplitude modulated signals, which are then directly detected. Differential detectors for binary differential phase shift keying (DPSK) and differential quadrature phase shift keying (DQPSK) are shown in Fig. 2.3.

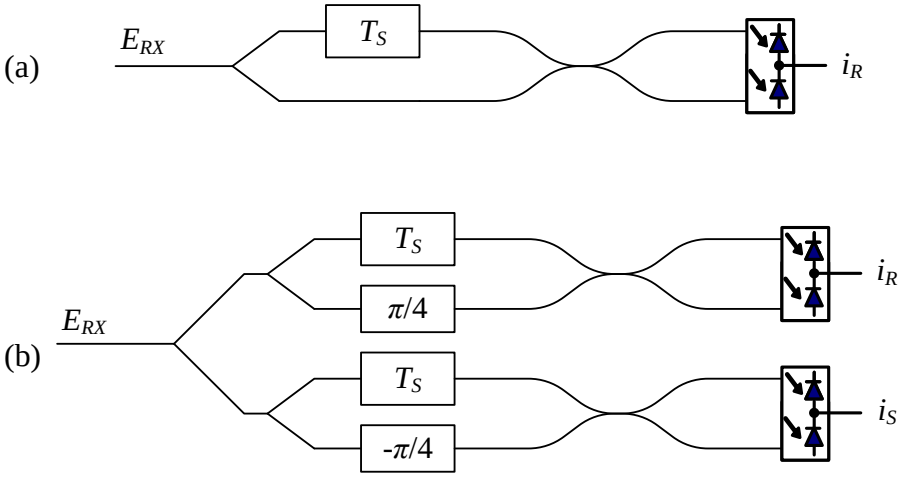


Fig. 2.3. Differential receiver architectures. BPSK (a) and QPSK (b)

A coherent receiver uses a local oscillator (LO) laser to mix with the received optical signal for intradyne or homodyne reception. (Here, homodyne refers to the use of the same laser source is used for transmit and receive LO, whereas an intradyne configuration uses independent lasers operating at approximately the same wavelength, with a frequency difference much less than the symbol rate). The advantage of differential detection is in complexity; no laser is required at the receiver, and the output of a differential detector requires only a slicer and clock recovery circuit. On the other hand, the differential detector limits the amount of digital signal processing available to the receiver since the signal has already undergone an irreversable nonlinear transformation (through the interferometer, square-law photodiode, and limiting amplifier). This implies that compensation for linear channel effects (e.g. dispersion and polarization demultiplexing) must be performed on the optical signal prior to detection.

As an alternative to differential detection, coherent detection provides higher performance at the expense of implementation complexity. The coherent optical receiver (Fig. 2.4) includes a laser LO, which for typical applications is independently tuned to a

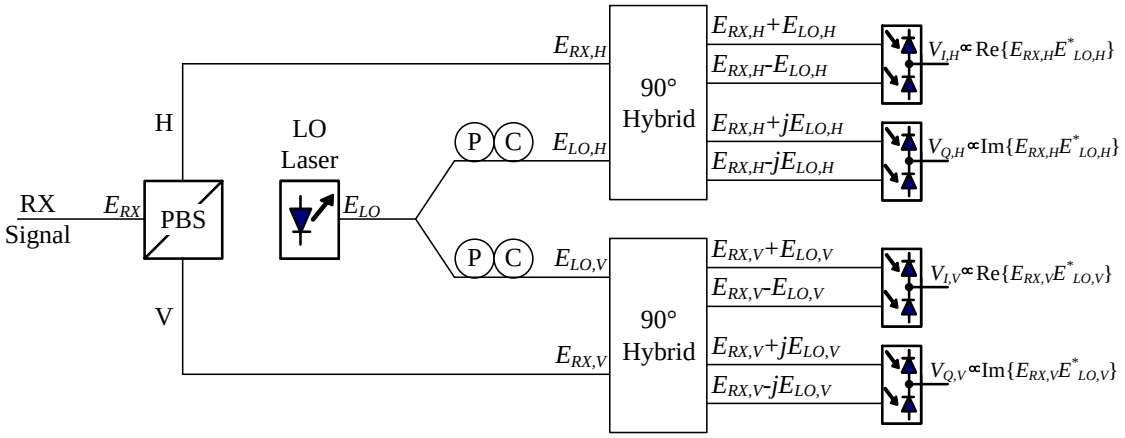


Fig. 2.4. Coherent optical receiver, with I&Q photocurrent outputs for each of two arbitrarily aligned linear polarization modes

wavelength near the center of the desired channel (intradyne). A polarization beam splitter (PBS) extracts two orthogonal linear polarization modes from the incoming optical signal. Each polarization mode enters a 90° optical hybrid alongside the LO output (aligned to each polarization correspondingly). The transfer function of an ideal two-input, four-output hybrid device is

$$\begin{bmatrix} E_{Out\,1} \\ E_{Out\,2} \\ E_{Out\,3} \\ E_{Out\,4} \end{bmatrix} = (1/2)\begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & j \\ 1 & -j \end{bmatrix}\begin{bmatrix} E_{In\,1} \\ E_{In\,2} \end{bmatrix} \tag{2.28}$$

The receiver uses balanced photodetectors, each producing output current proportional to the square of the incident electric field:

$$i_{OUT,IP} \propto \left|E_{RX} + E_{LO}\right|^2 = \left|E_{RX}\right|^2 + 2\,\mathrm{Re}\left\{E_{RX}E^{*}_{LO}\right\} + \left|E_{LO}\right|^2 \tag{2.29}$$

$$i_{OUT,IN} \propto \left|E_{RX} - E_{LO}\right|^2 = \left|E_{RX}\right|^2 - 2\,\mathrm{Re}\left\{E_{RX}E^{*}_{LO}\right\} + \left|E_{LO}\right|^2 \tag{2.30}$$

$$i_{OUT,QP} \propto \left|E_{RX} + jE_{LO}\right|^2 = \left|E_{RX}\right|^2 + 2\,\mathrm{Re}\left\{jE_{RX}E^{*}_{LO}\right\} + \left|E_{LO}\right|^2 \tag{2.31}$$

$$i_{OUT,QN} \propto \left|E_{RX} - jE_{LO}\right|^2 = \left|E_{RX}\right|^2 - 2\,\mathrm{Re}\left\{jE_{RX}E^{*}_{LO}\right\} + \left|E_{LO}\right|^2 \tag{2.32}$$

The balanced configuration cancels common-mode photocurrents, $i_{OUT,I} \propto \mathrm{Re}\{ E_{RX} E_{LO}^{*} \}$ and $i_{OUT,Q} \propto \mathrm{Im}\{ E_{RX} E_{LO}^{*} \}$.

The LO field is required to have stable phase and intensity such that the term E_{LO}^{*} becomes a constant scalar in (2.29-2.32), thus the currents vary proportionally to the real part (in-phase, or I) and imaginary part (quadrature, or Q) of the received signal. The coherent optical receiver is therefore capable of detecting the complex envelope of two orthogonal linear polarization modes, albeit with arbitrary orientation with respect to the transmitted polarization modes. Skew between the optical and electrical signal paths, common mode rejection ratio (CMRR), and gain imbalance are all key performance metrics of a coherent receiver.

2.4 Recent Research: 100 Gb/s Optical Transmission

The first major 100 Gb/s per wavelength systems to be demonstrated and deployed are based on DQPSK transmission. Winzer, et. al. demonstrated a single-polarization DQPSK system on a 100 GHz grid [6]. Other experiments, including those in the 100G Consortium lab, have successfully demonstrated DQPSK across two polarizations on a 50 GHz grid [7]. The low complexity of DQPSK as well as its reasonably high SE and error performance makes it an exceptional candidate for 100 Gb/s deployment.

Other efforts have focused on the digital signal processing (DSP) capabilities enabled by a coherent optical receiver. Earlier work by Ip and Kahn [22] on digital polarization demultiplexing and dispersion compensation paved the way for coherent experiments. Fludger, et. al. [5] demonstrated coherent optical detection of 100 Gb/s POLMUX QPSK, with timing and carrier recovery in addition to polarization

demultiplexing and channel equalization. All initial experiments have been performed by first capturing data on a high-speed digital oscilloscope, then exporting the data for processing on a computer. This method prohibits the use of mixed-signal circuitry for locking the LO to the received carrier or closing the timing recovery loop at the analog-to-digital converter (ADC) sample clock. Note that locking the LO is not anticipated, since the laser module linewidth and tuning control are likely to be set independent of the demodulator line card (regardless of other technical challenges of this approach).

Besides differential and coherent QPSK, other modulation formats are under study for 100 Gb/s optical links. OFDM under coherent optical reception has been demonstrated with encouraging results [23-24]. The OFDM format offers advantages in receiver complexity through block processing, frequency domain equalization, and the unique ability to separate sub-bands from the received envelope and process them in parallel. The disadvantage of OFDM is the well known problem of high peak-to-average power ratio (PAPR), which may make it more susceptible to nonlinear refraction than other linear modulation formats [25-27].

Another modulation format under consideration is multi-level minimum shift keying (ML-MSK). MSK is a widely used instance of full-response CPM with h=0.5, M=2, and $g(t)$ is a rectangular pulse (duration T_S) (parameters defined in Sec. 3.1). ML-MSK is the superposition of two synchronous MSK signals, each carrying independent binary data and each added at a different power level. ML-MSK with differential detection has been considered for fiber-optics [28], but does not compare favorably with DQPSK in terms of error performance.

One other field in optical research is the use of the split-step Fourier backpropagation method to solve the nonlinear Schroedinger equation that describes fiber propagation behavior [21]. This method enables the receiver to determine the SPM phase modulation term from its received data, and would possibly enable XPM correction if all active channels terminate at the same point. Additionally, the processing required for a simple system is very burdensome, and not yet feasible at these symbol rates.

The ultimate goal for future fiber-optic communication links is to maximize data rate at the lowest possible cost per bit. Other communication channels (wireline and wireless) rely on high SE over relatively shorter spans to achieve this goal, whereas optical communication channels use the fiber media to enable simpler schemes at much higher data rates. The capacity of the fiber channel is vast, though practical considerations require the use of WDM to divide it into sub-channels, each with its own laser source, modulator, ROADM filter, receiver, etc. Recent efforts have quantified the per-channel capacity of WDM systems (summarized in [21]), which is maximized between the ASE noise dominated OSNR limitation at low launch powers and the nonlinear phase noise dominated behavior at high launch powers. This concept of the finite and calculable capacity of nonlinear fiber is central to the motivation of this effort.

2.5 Georgia Tech 100G Consortium Research

Founded by Prof. Stephen Ralph in 2008, the Georgia Tech 100G Consortium is a partnership between multiple GA Tech faculty, students, and industry sponsors. The mission of the 100G Consortium is to explore optical, electronic, and signal processing interactions and develop algorithms, design rules, and technology for the deployment of 100 Gb/s (per lambda) networks. The 100G Consortium is equipped for simulation and

experimentation of several aspects of fiber-optic links. Simulation tools include the RSoft® Optsim™ optical simulator, which simulates electronic and optical components as well as signal propagation through fiber, and Matlab™ for signal processing within the transmitter and receiver paths. The experimental testbed comprises a vast amount of optical network equipment; lasers, electronic signal drivers, modulators, bit error rate testers (BERTs), fiber (SSMF, NZDF, and other types), EDFAs, wavelength selective switches (WSS), optical filters and attenuators, differential optical receivers, trans-impedance amplifiers (TIAs), photodiodes, coherent optical receivers, sampling and real-time oscilloscopes, optical dispersion compensation modules, and computer equipment to control much of the equipment.

Experimental efforts within the 100G Consortium have focused on establishing network deployment and scaling metrics, exploring different modulation formats, evaluating and isolating the effects of different components, and developing DSP algorithms for coherent demodulation. Early experiments assessed dispersion map strategies for deployments of 100G links alongside 10G channels within existing network topologies [29]. The consortium later gained access to a coherent optical receiver and real-time digitizing oscilloscope, expanding the work to include coherent reception [30]. Results from these studies indicate that best performance can be achieved with a small amount of residual dispersion per span (roughly 10%) along with a moderate amount of negative pre-compensation. In conjunction with these experiments, the RSoft simulation environment was carefully modified to achieve close matching to the experimental testbed, with a goal to achieve absolute matching (not relative) of BER vs. OSNR [31].

Subsequent experiments carried out on the testbed were to determine scaling behaviors of 100 Gb/s WDM deployments. The first was to develop a non-linear threshold (NLT) metric as a rule-of-thumb for predicting performance for scaling of links with a mix of 10 Gb/s OOK and 100 Gb/s QPSK carriers [32]. With coherent reception, all-digital dispersion compensation was compared to optical dispersion compensation, with the conclusion that the NLT makes a much better prediction for performance scaling in links without optical dispersion compensation than for links with in-line compensation. Another successful experiment was conducted to identify and predict the crosstalk penalty imposed by multiple adds and drops of a 100 Gb/s carrier within a WDM channel [33-34]. Not only did this result in the creation of a new method to quantify the in-band crosstalk penalty, but also identified an enhancement of the nonlinearity penalty in addition to the degradation due to nonlinearity without crosstalk.

Current efforts in the experimental testbed are to study techniques that will further identify scaling rules over new fiber types, and scaling data rates beyond 100 Gb/s. One observation for scaling higher than 100 Gb/s is the availability of QPSK modulators will be relatively cost-effective compared to other more sophisticated transmitter structures. Therefore the Consortium is interested in maximizing performance over closer channel spacing than the conventional 50 GHz grid. Experimentally, it was found that the channel spacing could roughly match the optical ROADM bandwidth without abrupt degradation, and without modification to any components in the system [35]. However, even closer spacing can be made feasible with advanced interference cancellation via DSP, which is presently under development. Also currently underway is a performance

study of different formats within a recirculating loop of a variety of fiber types, which enables significantly longer link distances.

In support of the Consortium's research described above, the DSP algorithms for coherent demodulation were developed in conjunction with the goals of this dissertation and are therefore described in detail in Chapter 4. Furthermore, the demodulation code in its entirety is contained in Appendix A. Simulations and experiments specific to research of modulation formats and demodulation algorithms are additional to those described above, and represent the major body of work represented by this dissertation. Hence these efforts are covered in detail in the following chapters.

CHAPTER 3

CONTINUOUS PHASE MODULATION

CPM is distinguished from other formats by its continuously constant amplitude, and as such it offers an interesting case for mitigating nonlinear refraction effects in fiber. A CPM signal is defined as [36]

$$s(t;\boldsymbol{\alpha}) = \sqrt{\frac{2E_S}{T_S}} \cos\left(2\pi f_c t + \varphi(t;\boldsymbol{\alpha})\right) \tag{3.1}$$

where the carrier phase $\varphi(t;\boldsymbol{\alpha})$ is the modulated phase, and is related to the data symbols through the following relationship:

$$\varphi(t;\boldsymbol{\alpha}) = 2\pi \sum_{k=-\infty}^{n} \alpha_k h_k q(t - kT_S) \tag{3.2}$$

.

Here, $\{\alpha_k\}$ are the transmitted symbols, chosen from an alphabet of ±1, ±3, ... ±(M-1) for an M-ary alphabet. Due to the summation relationship of $\varphi(t;\boldsymbol{\alpha})$ over all previously transmitted symbols, CPM is categorized as a modulation format *with memory*. The modulation index h_k determines the extent of the phase transition between symbols, and as such is tied intrinsically to both SE and normalized minimum Euclidean distance (hereafter referred to as *minimum distance*) between adjacent symbols. In general, h_k can vary from symbol to symbol in a cyclic manner (multi-h CPM); with careful selection it is possible to achieve higher minimum distance without reducing spectral efficiency. However, a fixed value is generally preferred for complexity concerns.

The pulse window function $q(t)$ is represented as the integral of a pulse shape $g(t)$:

$$q(t) = \int_0^t g(\tau)\,d\tau, \quad \begin{matrix} g(t<0) = 0 \\ g(t > LT_s) = 0 \end{matrix} \tag{3.3}$$

Cases for which $g(t)$ is non-zero over a single symbol (L=1) are known as *full response* CPM, and those for which it spans multiple symbols (L>1) are *partial response*. One common pulse shape is the rectangular pulse (REC) where $g(t)$ is constant over the span:

$$g(t) = \begin{cases} \frac{1}{2LT} & 0 \le t \le LT \\ 0 & otherwise \end{cases} \tag{3.4}$$

The REC pulse offers the lowest possible frequency deviation throughout the course of the phase transition, providing excellent SE for the full response case.

The temporal raised-cosine pulse (RC) pulse is another alternative:

$$g(t) = \begin{cases} \frac{1}{2LT}\left[1 - \cos\left(\frac{2\pi t}{LT}\right)\right] & 0 \le t \le LT \\ 0 & otherwise \end{cases} \tag{3.5}$$

which offers the most gradual change in frequency throughout the transition. The RC pulse is noted for its sharp rolloff when used in a partial response system. Other pulse shapes are possible, though the REC and RC are boundary examples for minimal frequency offset and minimal instantaneous change in frequency.

Selection of the parameters h, M, L, the pulse shape $g(t)$, and the number of symbol periods N observed by the receiver prior to detection altogether determine the SE, minimum distance, and implementation complexity of a CPM system. SE is highly dependent upon $g(t)$, which determines the rate of phase change, and thereby side-lobe rolloff characteristics. Note that full response signals have significantly more out-of-band energy than partial response ones, since partial response signals introduce intentional inter-symbol interference (ISI) to avoid abrupt phase changes. The common

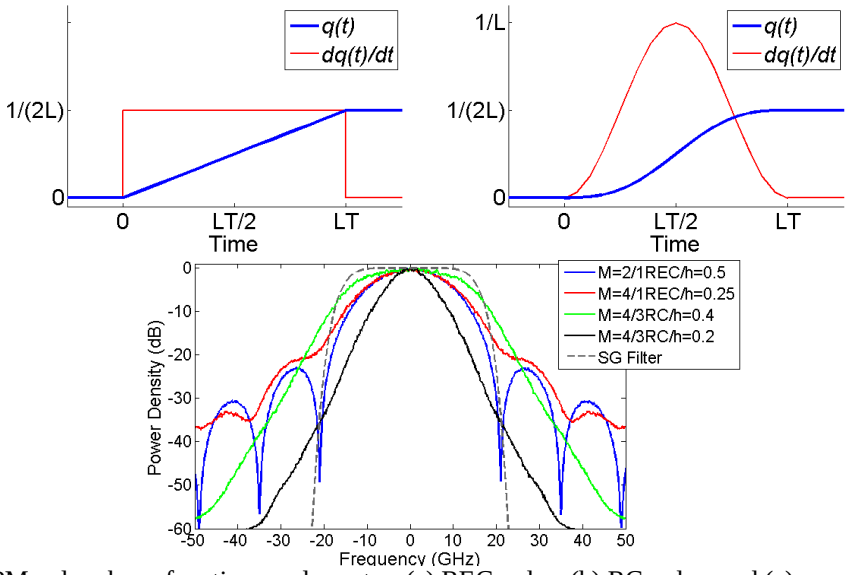


Fig. 3.1. CPM pulse shape functions and spectra: (a) REC pulse, (b) RC pulse, and (c) power spectral densities of a variety of CPM formats, plotted with a 40 GHz super-Gaussian bandpass filter (order 3.5)

REC and RC pulse shapes are illustrated in Figs. 3.1a and 3.1b, respectively. Spectral efficiency also depends highly on h and M, since the maximum frequency offset produced is proportional to $h(M\text{-}1)$, also impacting the width of the main spectral lobe. The power spectral densities of several CPM schemes are plotted in Fig. 3.1c, highlighting the impact of these parameters.

3.1 CPM Characteristics

Since CPM encompasses a broad range of parameters, it is valuable to introduce methods to assist in parameters that are well suited for an application. These tools include the *phase tree* and *phase difference tree*, and will be instrumental in determining the minimum distance of a CPM scheme. The phase of a CPM signal may be written with respect to a phase state description as follows:

$$\begin{aligned}\varphi(t;a_k) &= 2\pi\sum_{k=0}^{n} a_k h_k q(t-kT_S) \\ &= \pi h\sum_{k=0}^{n-L} a_k + 2\pi h\sum_{k=n-L+1}^{n} a_k q(t-kT_S) \\ &= \Phi_n + \Phi(t;a_k),\ nT_S \le t \le (n+1)T_S\end{aligned} \tag{3.6}$$

where Φ_n represents the state phase at $t = nT_S$.

For a full response system whose modulation index is rational (i.e. $h=m/p$), there are p different phase states with values

$$\Phi_n \in \left\{0, \frac{\pi m}{p}, \frac{2\pi m}{p}, ..., \frac{(p-1)\pi m}{p}\right\} \tag{3.7}$$

when m is even, and $2p$ states with values

$$\Phi_n \in \left\{0, \frac{\pi m}{2p}, \frac{2\pi m}{2p}, ..., \frac{(2p-1)\pi m}{2p}\right\} \tag{3.8}$$

when m is odd. The number of phase states for a partial response ($L>1$) is equal to pM^{L-1} when m is even and $2pM^{L-1}$ when m is odd.

Using (3.6), a phase tree is constructed by sketching the ensemble of all possible phase trajectories. The phase tree is an instructive tool to determine basic properties of a CPM signal. The phase trees of a binary full response 1REC and binary partial response 3REC system are sketched in Fig. 3.2a and Fig. 3.2b, respectively, both with modulation index $h=½$. In both cases, the signal is observed over $N=4$ interval symbols. As shown in Fig. 3.2b, a higher number of phase states are observed with the partial response as well as smoother transition between states. An important distinction, however, is that the maximum frequency offset in both cases is identical and depends only on the choice of pulse shape and modulation index. Therefore, the gain in spectral efficiency of the partial

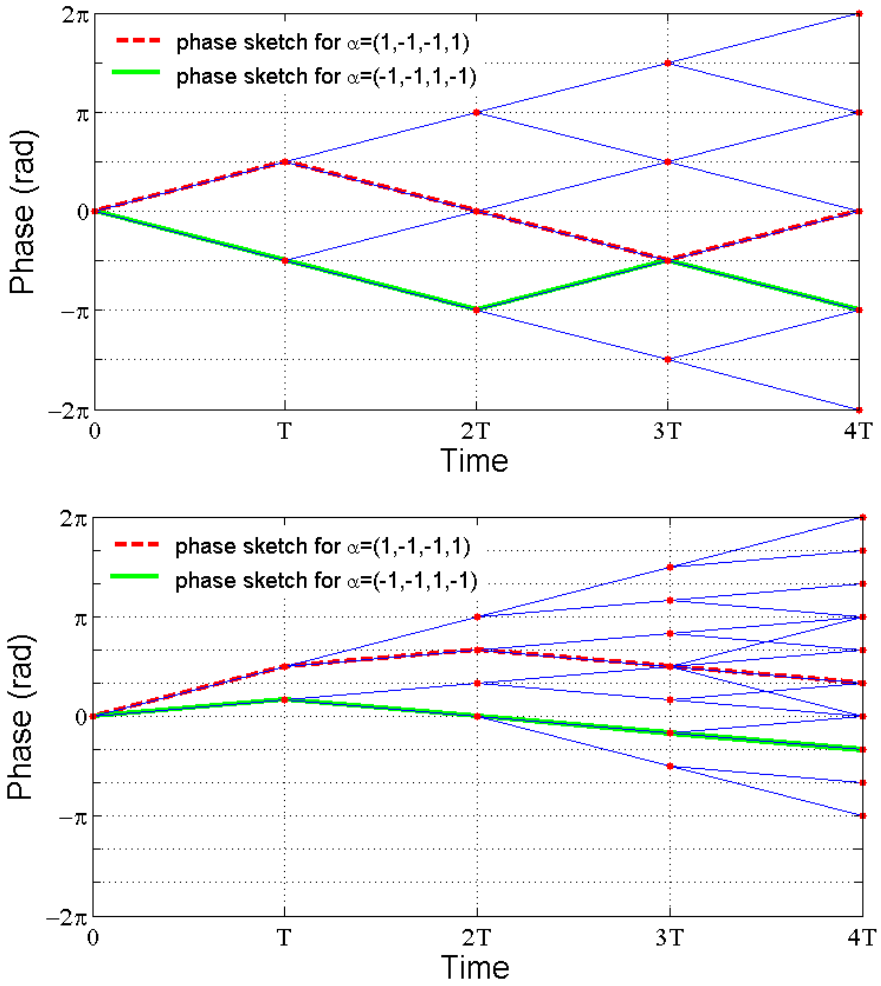


Fig. 3.2. Phase tree of binary CPM schemes: (a) full response 1REC, h=½, (b) partial response, 3REC, h=½.

response format is due only to the reduced probability of reaching the maximum frequency offset, not due to narrower range of frequency offsets.

The phase difference tree expands on the concept of the phase tree, where only the possible *differences* in the phase states and phase trajectories are plotted. The difference tree reduces the total number of possible paths that a pair of symbol sequences can span. Fig. 3.3 illustrates the phase difference path that results from a difference sequence of $\boldsymbol{\gamma}$=(2,0,-2,2), which corresponds to the pair of paths taken in Fig. 3.2 ($\boldsymbol{\alpha}_0$={1,-1,-1,1} and $\boldsymbol{\alpha}_1$={-1,-1,1,-1}). However, this difference sequence also covers the pair $\boldsymbol{\alpha}_0$={1, 1,-1,1} and $\boldsymbol{\alpha}_1$={-1,1,1,-1}, so the difference tree is preferable to use for minimum

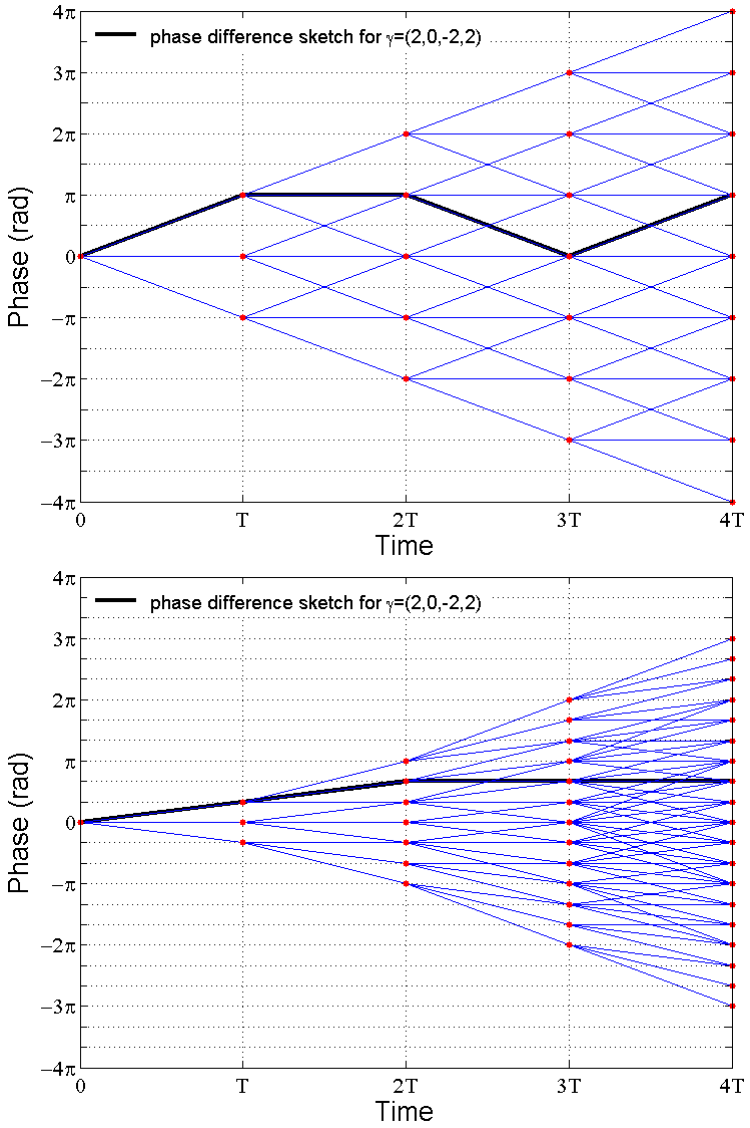


Fig. 3.3. Phase difference tree of binary CPM schemes (a) full response, 1REC, h=½. (b) partial response, 3REC, h=½.

distance calculation, as it reduces the number of possible sequences to $(M\text{-}1)\cdot(2M\text{-}1)^{N\text{-}1}$, rather than $(M\text{-}1)\cdot M^{2N}$ as defined by two independent phase trees.

3.1.1 Minimum distance computation

Minimum distance of a CPM signal can be computed by analyzing the range of phase difference sequences. Minimum distance for a digital modulation scheme is generally defined over the entire range of possible symbol sequences $\boldsymbol{\alpha}$ over the observation interval of N symbols:

$$d_{\min}^2 = \log_2(M) \cdot \min_{a_{k,N} \neq a_{l,N}} \left\{ \int_0^{NT} \left| s(t, a_{k,N}) - s(t, a_{l,N}) \right|^2 dt \right\}. \tag{3.9}$$

By inserting (3.1) into (3.9), the expression for minimum distance of a CPM signal over all possible difference sequences $\boldsymbol{\gamma}$ is simplified:

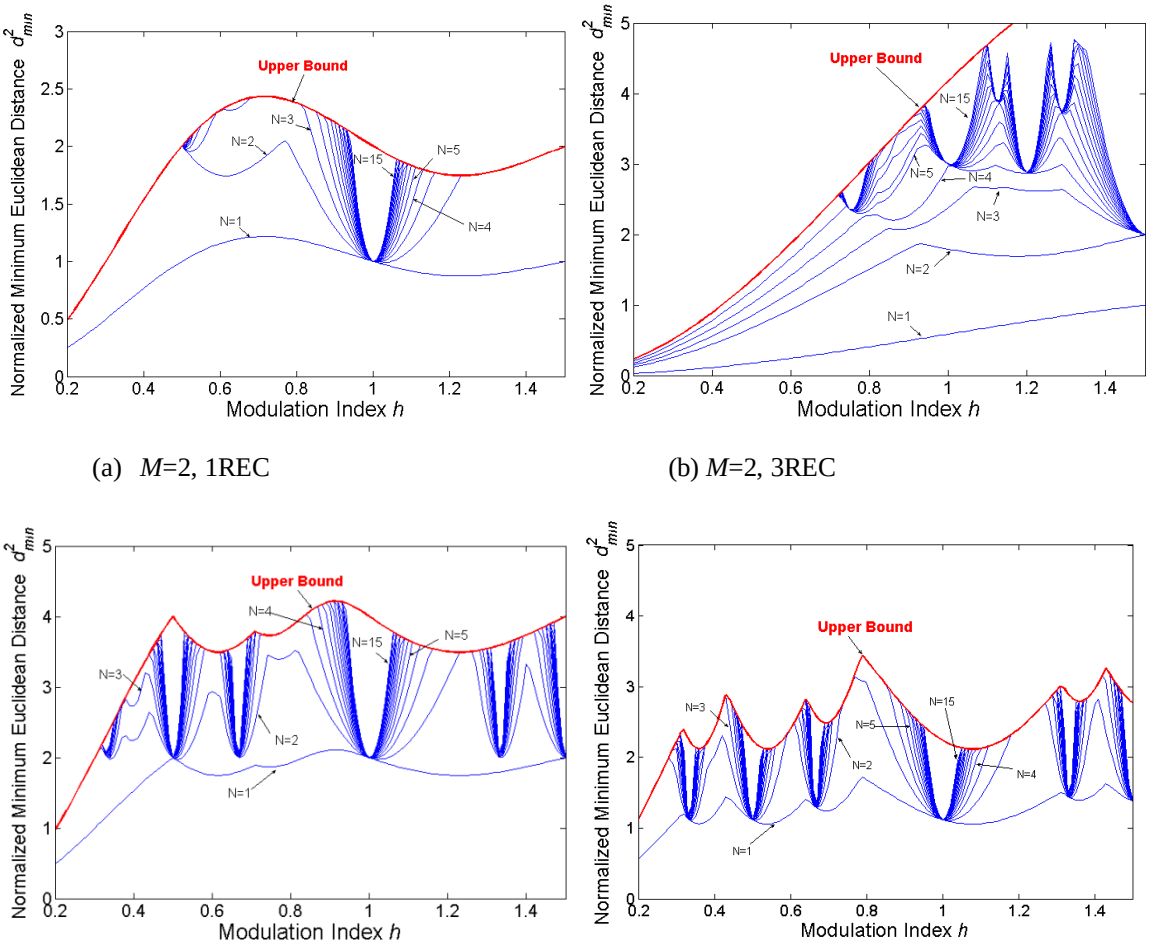
$$d_{CPM}^2 = \log_2(M) \cdot \min_{\gamma_N} \left\{ N - \frac{1}{T} \int_0^{NT} \cos\left[\varphi(t, \gamma_N)\right] dt \right\}$$
$$\gamma_0 = 2, 4, 6, \ldots, 2(M-1)$$
$$\gamma_{l>0} = 0, \pm 2, \pm 4, \ldots, \pm 2(M-1), \tag{3.10}$$

In this expression, minimum distance is a non-decreasing function of *N*, which is intuitive since it is natural to expect that as the receiver observes more symbols in its estimation of the initial symbol in the sequence it can only improve upon the initial estimate. However, there is a limit to the effectiveness of extending the receiver observation interval, which depends on the signal parameters (especially *L*). Note that the first symbol of the phase difference sequence γ_0 is non-zero, owing to the fact that the first symbol in the sequence must differ to establish minimum distance. Furthermore the symmetry of the cosine function allows the negative values of γ_0 to be ignored since each negative difference will have the same result as the corresponding positive difference.

The minimum distance is upper-bounded by the distance achieved between t=0 and the *first merge* point, where two symbol sequences that differ in the first symbol re-merge with the opposite difference in the second symbol. In other words, the first merge results from a phase difference sequence of $\boldsymbol{\gamma}=\{\gamma_0,-\gamma_0,0,0,\ldots 0\}$, under which constraint (3.9) yields the upper bound:

$$d_B^2(h) = \log_2(M) \cdot \min_{1 \le k \le M-1} \left\{ 2 - \frac{1}{T} \int_0^{2T} \cos\left[2\pi h \cdot 2k\left(q(t) - q(t-T)\right)\right] dt \right\}. \tag{3.11}$$

Whichever first merge yields the worse minimum distance will act as an upper boud to the CPM signal's minimum distance, but this may not be achievable based on overlap in the possible phase trajectories (depending on the signal parameters). Figures 3.4 and 3.5 illustrate the relationship between minimum distance, h, and N, for five CPM schemes. There are a few features that are common to all schemes. First, note that an increase in N never reduces minimum distance, though there is not always an improvement. Another feature is that the optimal receiver observation over the region $h=(0, {}^{1}/_{M})$ is achieved with only two symbols (N=2). The reason for this is that the phase



(a) M=2, 1REC

(b) M=2, 3REC

(c) M=4, 1REC

(d) M=4, 1RC

Fig. 3.4 Minimum distance vs. modulation index for CPM schemes: (a) binary 1REC, (b) binary 3REC, (c) 4-ary 1REC, and (d) 4-ary 1RC, with non-decreasing bands for increasing N.

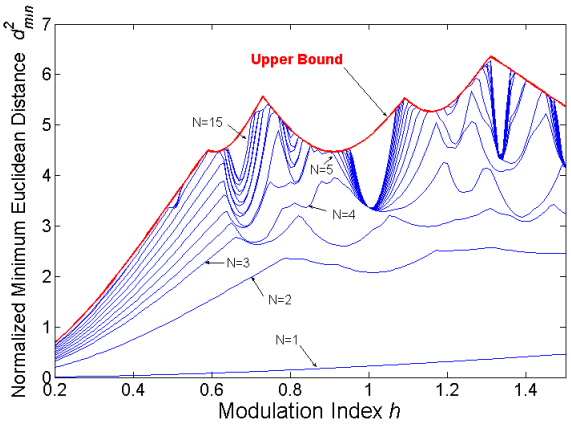


Fig. 3.5 Minimum distance vs. modulation index for 4-ary 3RC, shown over a variety of N.

never wraps (modulo 2π) over the first two symbols, so the minimum distance over that span is identical to the first merge point. Beyond this region, more than two symbols may be required to achieve the optimal minimum distance. A third feature depicted in Figs. 3.4 and 3.5 is the presence of weak modulation indexes; those for which the achievable minimum distance is at a local minima distant from the upper bound, and insensitive to increasing N. Examples of weak modulation indexes for M=4, 1REC CPM are $^1/_3$, $^1/_2$, $^2/_3$, 1, $^4/_3$, and $^3/_2$, and should usually be avoided in lieu of others that provide better minimum distance.

A fast sequential algorithm has been developed for the purpose of quickly computing the minimum distance of any CPM scheme [37]. To converge rapidly, the algorithm makes use of the constraints that minimum distance is non-decreasing with N as well as the first merge upper bound in order to eliminate sets of symbol sequences. The algorithm iterates over γ and N for each value of h analyzed, using (3.10). A flow diagram of the fast sequential algorithm is shown in Fig. 3.6 [37].

First, the upper bound is computed to be used as a discard threshold, and initial values are established. N initializes to 1 (single symbol observation), h initializes to the

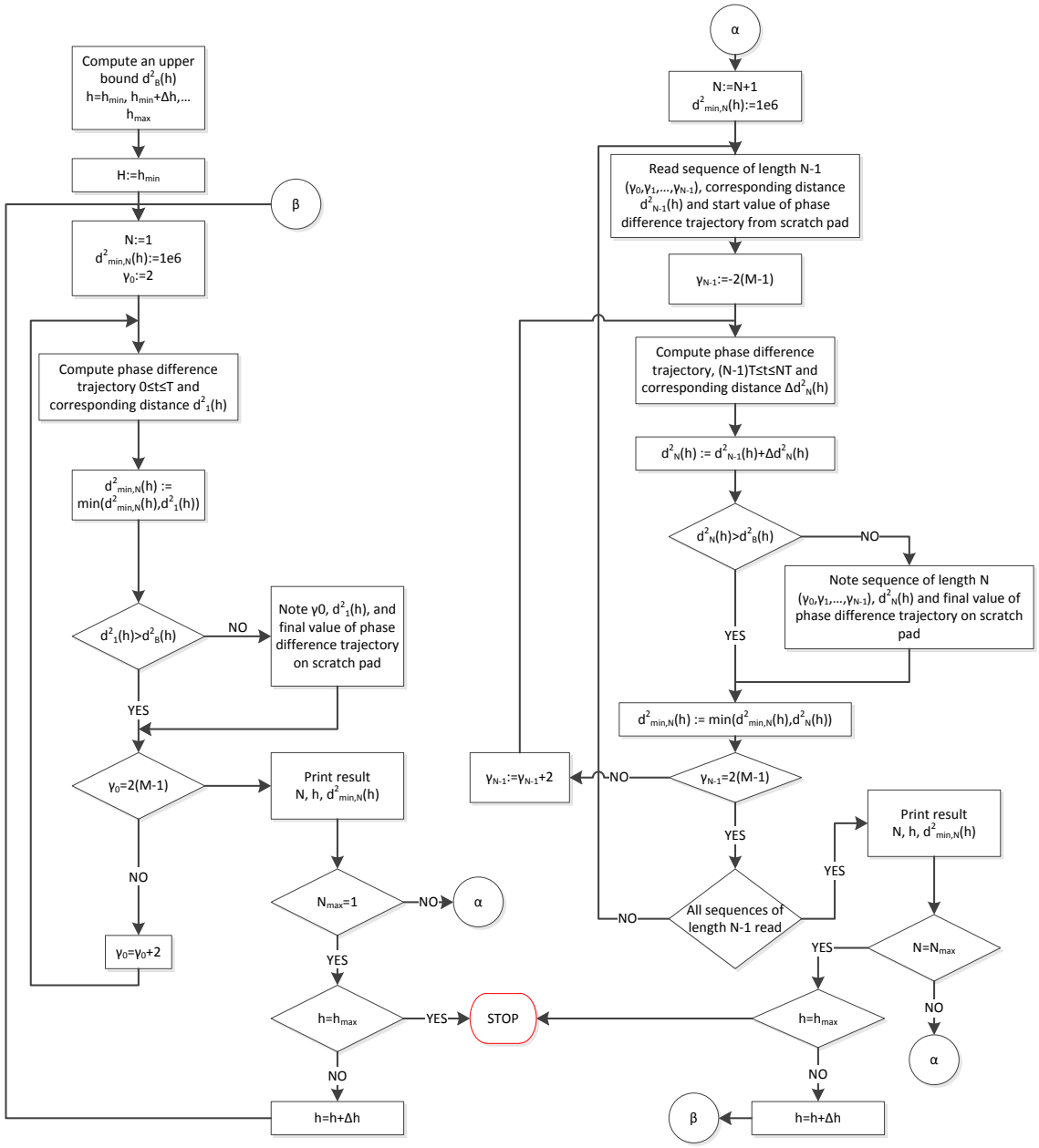


Fig. 3.6. Fast sequential algorithm flowchart [37]

h_{min} (the minimum of the desired range), d_{min} initializes to a value much greater than the upper bound, and γ_0 is set to 2, which is the minimum phase difference between the first two symbols. The algorithm then iterates through γ, recording the minimum distance and for each evaluated sequence. Whenever a sequence exceeds the upper bound, it is removed from the available set of sequences for subsequent iterations through N. Thus,

the set of sequences is maintained well below the full set of possible difference sequences $(M\text{-}1)\cdot(2M\text{-}1)^{N-1}$ and the algorithm converges quickly.

3.2 CPM Signal Generation and Reception

Any signal modulated for transmission over a channel with bandwidth B can be generated by a pair of DACs controlling a quadrature modulator, provided the sampling rate exceeds B. However, the resolution of the DACs must also be considered carefully in an optical application since the channel bandwidth can exceed 40 GHz. One interesting feature of CPM is that the phase may grow (or diminish) unbounded, therefore a phase modulator alone is not sufficient to produce the signal unless it is capable of wrapping from 0 to 2π without a jump-discontinuity. An electro-optic phase modulator does not provide this capability, so an alternative method of signal generation is required. CPM signal generation in fiber is discussed in detail in Ch. 4.

The implementation of an optimal CPM receiver is complex as it requires, in general, the use of a bank of matched filters specific to the modulation format [36]. The filters are dependent on the sequence of transmitted symbols over a chosen symbol observation interval, and also on the parameters of the scheme h, M, L, and $g(t)$. It is highly desirable to implement any required filter bank digitally since the corresponding analog or optical complexity would likely be prohibitively complex, and would not be flexible should any parameter change. Additionally, the optimal receiver output is best processed by maximum likelihood sequence detection (MLSD), which implies the use of the Viterbi algorithm. The choice of receiver implementation is very important in determining the usefulness of any CPM scheme, especially in the high speed optical application.

CPM is inherently a nonlinear modulation format, since the transmitted signal is not a linear combination of the symbols, as evident in (3.1) and (3.2). Necessary receiver functions, such as equalization, timing recovery and carrier phase recovery, are burdened by this nonlinear characterization. In 1986 Laurent published a method of decomposing any binary CPM format into a set of pulse amplitude modulation (PAM) waveforms, multiplied by data-dependent complex coefficients and superimposed on a symbol-by-symbol basis [38]. Later work [39] extended the Laurent decomposition to *M*-ary CPM schemes for $M=2^P$ by recognizing that any such scheme is the product of *P* binary schemes. A description of the linear decomposition for full-response *M*-ary CPM, as well as optical CPM receiver design, is provided in Ch. 4 and Appendix B. Note that the linear description can be utilized for the transmitter or receiver implementation independently. In other words, the transmitter need not be implemented linearly to match a receiver based on the linear CPM decomposition.

3.3 Application to Fiber-Optics

The constant envelope feature of CPM and its SE advantage over formats such as frequency shift keying are the compelling reasons for its use. Typically CPM has been deployed in satellite and mobile wireless communication systems, where constant envelope enables the use of nonlinear radio front-ends with high power efficiency and low cost. In fiber, the constant envelope feature does not reduce the cost or power of the transmitter, but it does limit the fluctuation in intensity of the signal. Therefore, a single CPM channel does not produce much SPM in fiber. However, dispersion in fiber will distort the signal and eventually degrade its constant intensity. Additionally, variation in

optical intensity will arise whenever multiple modulated signals are optically combined to propagate through a link of fiber.

CPM proponents often describe it as a format with high power efficiency and spectral efficiency [36-39], though this statement does not hold in comparison to filtered QPSK and higher order quadrature amplitude modulation (QAM) formats. QAM symbols are placed at equidistant square gridpoints in the I/Q plane, and therefore are described with two degrees of freedom (viz. I and Q in Cartesian coordinates, or amplitude and phase in polar coordinates). CPM, like PSK, is restricted so that all symbols lie at the same radius on the I/Q plane, yielding only one degree of freedom. As the symbol alphabet for CPM increases, the minimum distance between symbols decreases more rapidly than QAM due to this restriction. Therefore, the use of CPM in optical systems is limited to constellations with modest SE and is not predicted to scale well above M=8. Furthermore, nonlinear impairments in fiber manifest as phase noise, attacking the only degree of freedom distinguishing CPM symbols.

The Optical Internetworking Forum and Ethernet Task Force bodies have selected POLMUX DQPSK as the format for initial 100 Gb/s standards. POLMUX DQPSK will therefore be a key benchmark to which all other modulation formats shall be compared, including this CPM study in particular. One key objective is to identify the modulation format parameters and fiber link configurations, if any, which produce favorable results for CPM over coherently demodulated DQPSK. Furthermore, consideration will be made to other formats that exhibit similar properties to CPM but that can achieve gain over QPSK without the burden of excess implementation complexity.

3.4 Alternative Linear Formats

Though CPM provides the desirable constant intensity feature, this constraint leads to higher implementation complexity than memoryless linear formats like QPSK. QPSK exhibits 100% fluctuation in intensity, though alterations exist to reduce the extent of fluctuation while maintaining simplicity, SE, and performance. One such alteration is offset QPSK (OQPSK), wherein the I and Q components are delayed by $T_S/2$ with respect to each other. As a result, either component reaches its maximum amplitude during the zero-crossing of the other, i.e. the complex envelope never passes through the origin of the I/Q plane. The other case is correlative PSK, where the symbols lie on the unit circle on the I/Q plane, but symbol transitions that pass through low intensity regimes are prohibited. One example based on QPSK is $\pi/4$-shifted QPSK, which alternates between the two distinct QPSK sub-constellations (separated by $\pi/4$ phase) that together make up the 8-PSK alphabet. Assuming temporal raised cosine pulses (a common approximation for optical linear modulation pulses), the range of intensity fluctuation of OQPSK is ~3.3 dB, and is ~8.3 dB for $\pi/4$-shifted QPSK (constellations in Fig. 3.7).

OQPSK and $\pi/4$-shifted QPSK are linear alternatives for reduced intensity fluctuation which are intended to avoid exciting nonlinear refraction. On the other hand, it is possible to utilize a linear modulation format that is robust to the phase noise generated by nonlinear refraction. Star-QAM constellations fall into this category, or in general any format that places symbols on concentric circles in the I/Q plane. This is a departure from the normal square QAM constellations, whose outer constellation points would dominate the error performance as the phase noise tolerance would be correspondingly narrower. This also implies that the distribution of symbols on the

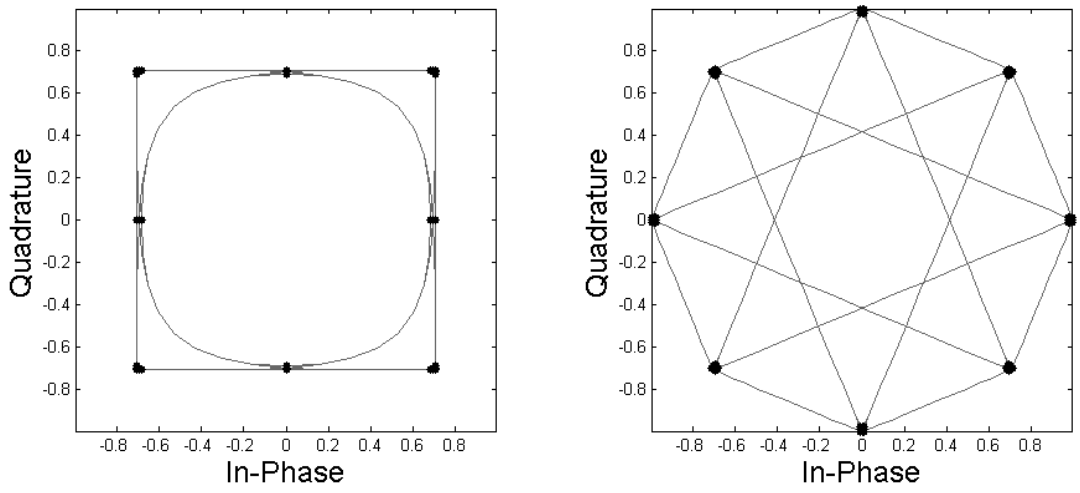


Fig. 3.7: OQPSK (a) and π/4-shifted QPSK (b) constellations, assuming temporal RC pulse shape

circles should be roughly constant as the circle radius increases. It has been shown that the use of such constellations is required to achieve SE near the nonlinear fiber capacity bound, assuming the channel capacity exceeds ~2.5 b/s/Hz [21, 40]. Importantly, CPM formats should be viewed as capacity-reaching for links below this threshold.

3.5 CPM Research Trends

CPM continues to be an active area of research, in part due to the desirability of its unique constant amplitude feature, and in part due to the difficulty in realizing optimal reception with low complexity. Recent research has brought some aspects of the OFDM receiver process, which possesses complexity advantages for equalization in the frequency domain [41-43]. One such study involving block-based CPM transmission in a 60 GHz wireless application [42] lends itself well to this effort for its low-complexity all-digital implementation. Block-based CPM combines the linear decomposition of the signal with an OFDM-style receiver. The result is an efficient design capable of equalizing channel effects, acting as a filter bank, and enabling Viterbi decoding in the time domain. Other recent advancements include improvements on timing and phase

recovery of the CPM signal [43-44], which are both essential for coherent optical demodulation. The scheme in [43] presents the use of cyclic prefix as an aid in synchronization, a strategy that has been used extensively in OFDM receivers [45]. Another relevant area of advancement is the concatenation of convolutional coding, low density parity check coding, and turbo coding techniques [46-48]. When used in conjunction with CPM, these inner codes drastically improve performance. However, it must be noted that high block lengths and extensive decoding iterations are particularly undesirable when dealing with symbol rates in excess of 25 Gbaud.

CHAPTER 4

COHERENT OPTICAL RECEPTION TECHNIQUES

The advent of the polarization-diverse coherent optical receiver (Fig. 2.4) enables a vast array of opportunities for DSP in the demodulation of optical signals. In this chapter, the set of algorithms used in a typical coherent optical receiver is explained. The challenge remains that the sample rate exceeds the available DSP operating frequency by over an order of magnitude. This is unlikely to change in the near future; that the electrical elements capable of operating at the required sample rate (or symbol rate) in CMOS will be limited to flip-flops separated by a few logic gates. Data conversion (i.e. DACs, ADCs) at these rates are likely to be fabricated in more exotic semiconductor substrates (e.g. InP, GaAs, SiGe). Therefore, systems should favor two implementation styles where possible:

1) feed-forward data paths (i.e. output states are not fed back for processing subsequent cycles), since they can be arbitrarily parallelized without performance penalty, though adding latency [49]

2) burst-mode receiver blocks for feedback paths, which operate over entire blocks of data and use the final states as initial conditions for processing a subsequent (though not contiguous) block.

With these considerations, the 100G Consortium's developmental demodulator code consists of the following stages (Fig. 4.1), the details of each are covered throughout

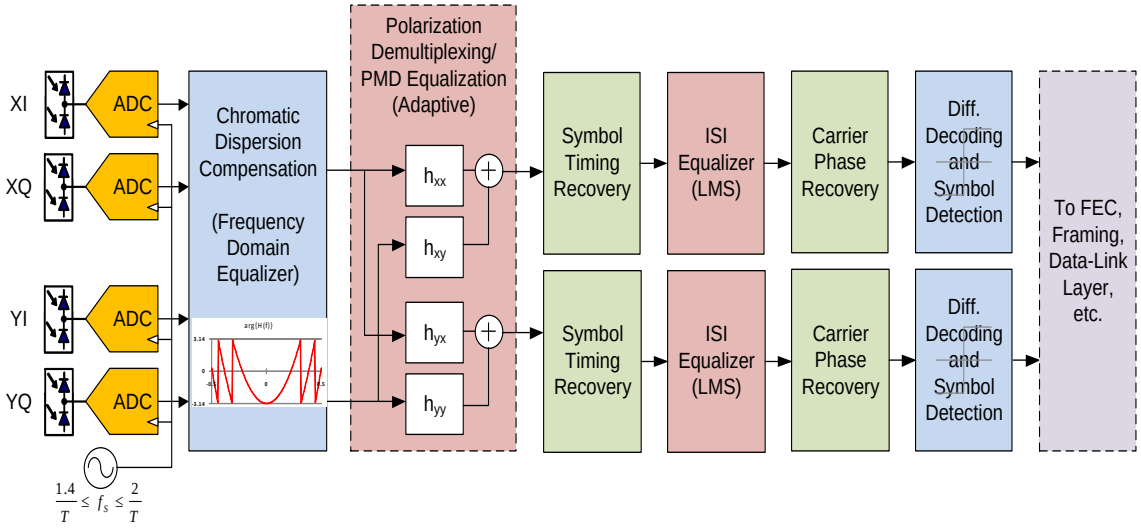


Fig. 4.1. Block diagram of 100G coherent receiver demodulator

the remainder of this chapter:

1) chromatic dispersion compenstation
2) polarization demultiplexing
3) symbol timing recovery
4) channel equalization (blind, decision directed, decision feedback)
5) carrier phase recovery

In addition to the above functions, Sec. 4.6 covers pseudo-random binary sequence (PRBS) pattern generation and matching and support for differential encoding. Not included in the demodulator is the enhanced FEC, which is presumed to be separated from the demodulator code as is typical for optical systems (though this relationship may evolve).

4.1 Chromatic Dispersion Compensation

An optical carrier entering the coherent receiver is optically processed to extract four photocurrents proportional to the I and Q components along two linear, orthogonal SOPs of the channel of interest. The photocurrents are digitized by four delay-matched

ADCs, all operating at the same sample rate, though asynchronous to the symbol rate. (A ratio of integers is preferred for the oversampling rate, at roughly 1.4 times the baud rate or greater). Immediately after digitization, the first operation performed is CD compensation. Several reasons support the placement of this unit up front. First, CD is independent of SOP and modulation format. Second, it is an all-pass transfer function, so there is (in principle) no frequency-selective attenuation that would require compensation by subsequent equalizers. Third, it reduces delay spread for the channel equalizers, making them more stable and faster to converge. Fourth, it avoids performance degradation among the timing recovery and polarization demultiplexing stages, both of which are sensitive to dispersion. And finally, CD is known *a priori* and is relatively fixed throughout time and operating conditions.

One straightforward method for CD compensation is to filter the signal with an FIR filter whose response is the all-pass inverse of (2.10):

$$H_{D,COMP}(\omega) = H_D^{-1}(\omega)\exp\{-j\delta\omega\} = \exp\left\{-j\left(\frac{\lambda_0^2 DL\,\omega^2}{4\pi c} + \delta\omega\right)\right\}, \tag{4.1}$$

where δ is a delay term to assure causality. The filter impulse response is $h_{D,COMP}$=IDFT{$H_{D,COMP}$}. The length of the impulse response (i.e. the number of taps with significant coefficients) goes linearly with increasing residual dispersion. Since it can be excessively long in certain cases, a frequency domain equalizer (FDE) implementation (also called "fast convolution") is desirable to reduce complexity [50].

The FDE simply incorporates (4.1) to compute the phase shift for each frequency bin. Again, the number of required bins is proportional to total dispersion. The desired output is

$$y[n] = x[n]*h_{D,COMP}[n], \tag{4.2}$$

where $x[n]$ is the discrete time input signal, and $h_{D,COMP}[n]$ is the CD compensation filter's impulse response, length L_h. Taking the DFT of both sides of (4.2) seemingly gives the result

$$Y(e^{j\omega})=X(e^{j\omega})\cdot H(e^{j\omega}), \tag{4.3}$$

however (4.3) performs a circular convolution of the finite-length sequences $x[n]$ and $h[n]$, whereas (4.2) is a linear convolution and the input signal $x[n]$ is indefinitely long.

A linear convolution of two finite length sequences (length L_x and L_h) can be achieved in the discrete frequency space if each is zero-padded to a matching length of $L_y \geq L_x+L_h-1$ samples:

$$\hat{x}_k = \begin{bmatrix} x[n + kL_x : n + (k+1)L_x - 1] \\ Z_x \end{bmatrix}, \tag{4.4}$$

$$\hat{h}_k = \begin{bmatrix} h[0 : L_h - 1] \\ Z_h \end{bmatrix}, \tag{4.5}$$

where Z_x and Z_h are column vectors of zeros of length L_y-L_x and L_y-L_h, respectively. Using an L_y-point FFT and multiplying the signals in the frequency domain (4.3), then transforming back to the time domain yields the linear convolution block $\hat{y}[n]$ (length L_y).

For a long $x[n]$ input (much longer than L_h), the sequence $y[n]$ can be extended over an infinite number of L_x-length blocks by adding the trailing L_y-L_x symbols of the previous block to the first L_y-L_x symbols of each block, i.e.:

$$y[n + kL_x : n + (k+1)L_x - 1] = \hat{y}_k[0 : L_x - 1] + \hat{y}_{k-1}[L_x + 1 : L_y - L_x - 1]. \tag{4.6}$$

This operation stitches the blocks together adding the zero-padded header and trailer of the $\hat{y}[n]$ blocks, a method known as overlap-add [50], and is illustrated in Fig. 4.2.

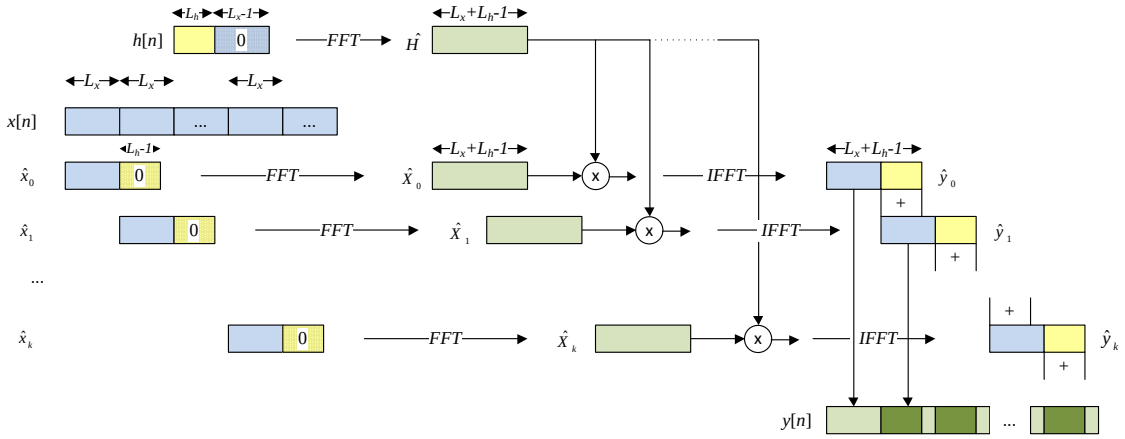


Fig. 4.2. Overlap-Add method for fast convolution, showing block assembly of output signal *y* by adding overlapping segments of circular convolution of zero-padded *x* and *h* inputs.

For simplicity, the 100G Consortium implementation zero-extends $h[n]$ and L_h length blocks of $x[n]$ to $2L_h$ samples. The first step of the algorithm is to pre-compute the FDE coefficients $\hat{H}[k]$. The L_{max}-point IFFT of the zero-delay, discrete frequency conversion of (4.1) is used to approximate $h[n]$ and its length L_h:

$$\tilde{h}[n] = IFFT\left(\tilde{H}[k]\right) = IFFT\left(\exp\left\{-j\left\lfloor\frac{\lambda_0^2 DL}{4\pi c}\left(\frac{\pi k}{L_{\max}} f_{samp}\right)^2\right\rfloor\right\}\right), k \in \left[-\frac{L_{\max}}{2}, \frac{L_{\max}}{2} - 1\right], \quad (4.7)$$

where L_{max} is arbitrarily large (e.g. 8,192). The resulting L_{max}-length impulse response $\tilde{h}[n]$ is truncated to L_h+1 taps (L_h is a power of two to minimize complexity) in which resides at least 99.99% of the coefficient power. This truncated impulse response, symmetric about its center tap, is then zero-padded to length $2L_h$ and denoted as $\hat{h}[n]$. Finally $\hat{h}[n]$ is passed through a $2L_h$-point FFT, and results in the static FDE coefficient block $\hat{H}[k]$.

With the pre-computed $\hat{H}[k]$ coefficients, the FDE is equalizes the infinite length $x[n]$ vector as follows. First, blocks of L_h samples of the $x[n]$ signal are zero-padded and

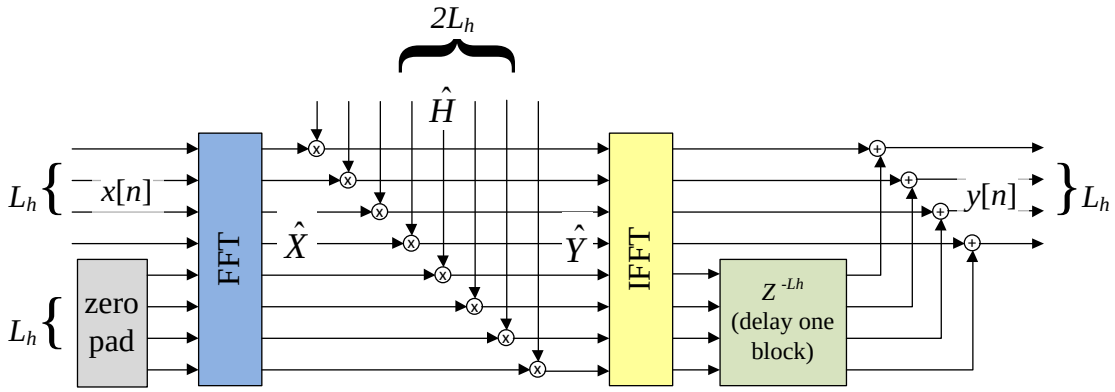


Fig. 4.3. Block diagram of FDE structure used for chromatic dispersion equalization, utilizing a *2Lh*-point FFT and IFFT for channel impulse response length L_h, and performing overlap-add at the output.

processed by a $2L_h$-point FFT to become $\hat{X}[k]$. Each element of $\hat{X}[k]$ and $\hat{H}[k]$ vecors are multiplied together, and the product is passed through a $2L_h$-point IFFT. The resulting $\hat{y}[n]$ blocks are merged with the overlap-add method.

Note that a time domain (transversal) equalizer requires L_h^2 complex multiplies for a block of L_h samples. The FDE requires $2L_h$ complex multiplies for the filter and $O(2L_h\log_2(2L_h))$ complex multiplies for the FFT and IFFT, which can be a significant reduction in complexity for systems with high residual compensation. Furthermore, several of the FFT/IFFT coefficients are simplified by use of a power of two L_h to values that do not require multiplication, such as ±1 and ±*j*. A block diagram of an FDE implementation for CD compensation is shown in Fig. 4.3; two such blocks are used to equalize the two received SOPs.

4.2 Polarization Mode Demultiplexing

Following CD compensation, the signal resembles the transmitted signal, though with additive noise, phase noise from nonlinearities and laser sources, and an unknown and time-varying SOP. At this stage it is typical for timing recovery to be performed in a PM-DQPSK system [5, 51], though performance of such an implementation may suffer if

the X and Y polarizations were launched with unaligned symbols at the transmitter. Instead, the 100G Consortium's demodulator demultiplexes the polarization modes at this stage. The objective of this module is to effectively estimate the fiber Jones matrix T (2.25) and invert it.

Polarization demultiplexing occurs here at the sample rate (asynchronously to the symbol rate) using a two-input, two-output adaptive equalizer operating with the constant modulus algorithm (CMA). The benefits of this novel approach are that it allows for arbitrary modulation format and arbitrary offset between X and Y polarization modes without suffering performance degradation [52]. The disadvantages are that overall system complexity may be increase, as ISI must still be adaptively equalized among each polarization mode. However, the alternative architecture incorporates a full two-input, two-output adaptive equalizer at twice the symbol rate, whereas this one operates at the lower ADC sample rate.

The CMA algorithm converges on signals without a training sequence, and is therefore known as a non-data aided (NDA) or "blind" method. The error function for CMA attempts to minimize, through steepest gradient approach, is:

$$e = y(R - |y|^2), \tag{4.8}$$

$$R = \frac{E\{|a_n|^4\}}{E\{|a_n|^2\}}. \tag{4.9}$$

Here y is the equalizer output, and R is a real-valued and positive radial constant. In other words, zero error occurs when the output signal is bound to a circle in the I/Q plane of radius $\sqrt{R}$. Though this algorithm is precisely suited for PSK systems (after timing recovery) and CPM system, its convergence is guaranteed for a complex-valued symbol alphabet a_n under the condition [53]:

$$E\{a_n^2\} = 0 \tag{4.10}$$

A sufficient criteria for meeting the convergence constraint of (4.10) is for the modulated signal to have rotational symmetry. In other words, there must be ambiguity upon any number of 90° phase shifts in the I/Q plane. This condition applies to QPSK, QAM, OQPSK, CPM, and PSK (except BPSK). Furthermore, an asynchronously sampled signal of those modulation formats is also rotationally symmetric (e.g. oversampling ratio of 10/7 in Fig. 4.4), though the step size should be small enough to average out the error

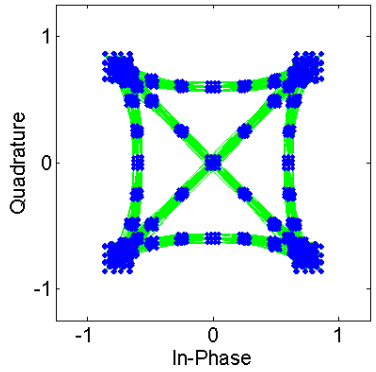


Fig. 4.4. Filtered RZ-QPSK signal in continuous time (green), and oversampled at 10/7≈1.43 samples per symbol (blue).

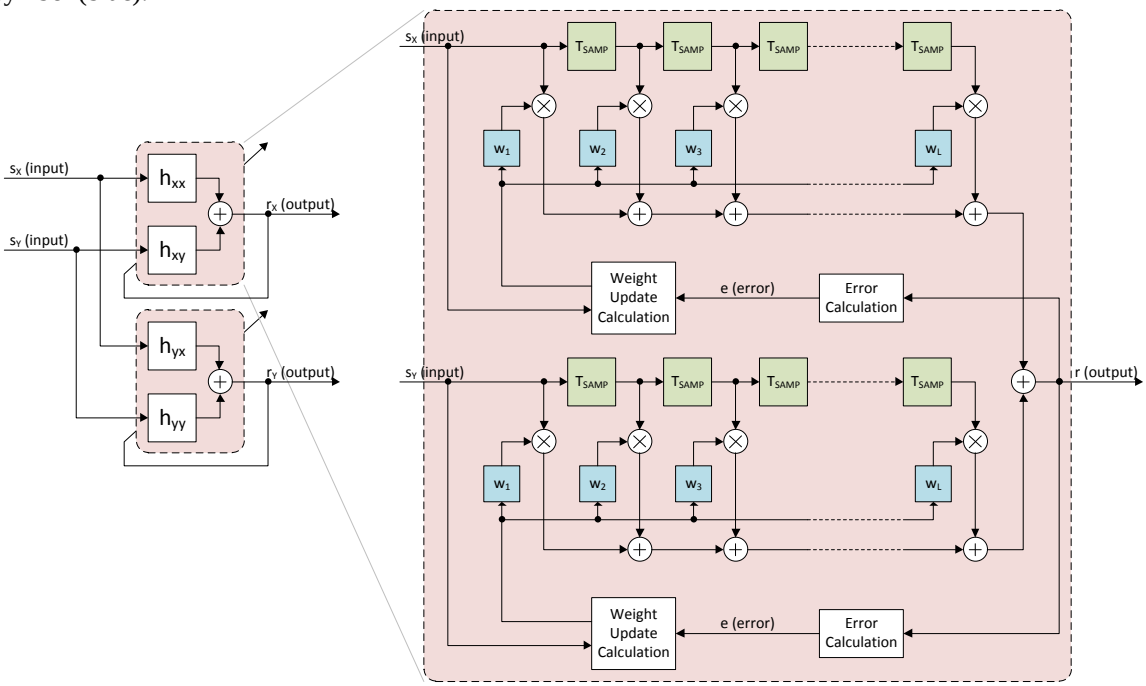


Fig. 4.5. Block diagram of 2x2 equalizer for polarization demultiplexing; entire structure on left, detail on right (one instance per polarization).

over a significant set of samples. Also, a low step size prevents instability of the channel ISI equalizer that follows. A block diagram of the blind asynchronous polarization demultiplexing equalizer is depicted in Fig. 4.5.

It is possible for the polarization demultiplexer based on the CMA algorithm to converge so that both outputs are locked to the same state of polarization. A method was devised [54] to avoid this degenerate behavior, by recognizing that the fiber Jones matrix (2.25) is constrained by orthogonality. More specifically, orthogonality is assured when the equalizer tap weights conform to:

$$\begin{bmatrix} \overline{h}_{xx} & \overline{h}_{xy} \\ \overline{h}_{yx} & \overline{h}_{yy} \end{bmatrix} = \begin{bmatrix} \overline{u} & \overline{v} \\ -\overline{v}^{*} & \overline{u}^{*} \end{bmatrix}, \tag{4.11}$$

where u and v are complex-valued vectors with one coefficient per equalizer tap. Note that complex conjugation of the coefficient vectors in (4.11) implies time reversal along the vector in addition to conjugation among each tap. In addition to this constraint, it is recommended to initialize the weight coefficients with a '1' in the center tap of h_{xx}, and zeros in all other locations to avoid the outputs converging to the same signal.

The polarization demultiplexer is designed to also detect the SOP of an incoming single-pol. transmission. In this mode of operation, the orthongonality constraint is set and the error vector for the Y output is simply $e = 0\text{-}y = -y$, which effectively minimizes the power on that output. Therefore, all signal power should be aligned to the X polarization output, and the Y output can be ignored.

For single-pol. or polarization multiplexed transmission, the unit can operate in burst mode provided that the SOP has not significantly changed between the end of the

previous block and the next block. Performance in burst mode may benefit from running the tail of the prior block as a preamble immediately prior the start of the next block.

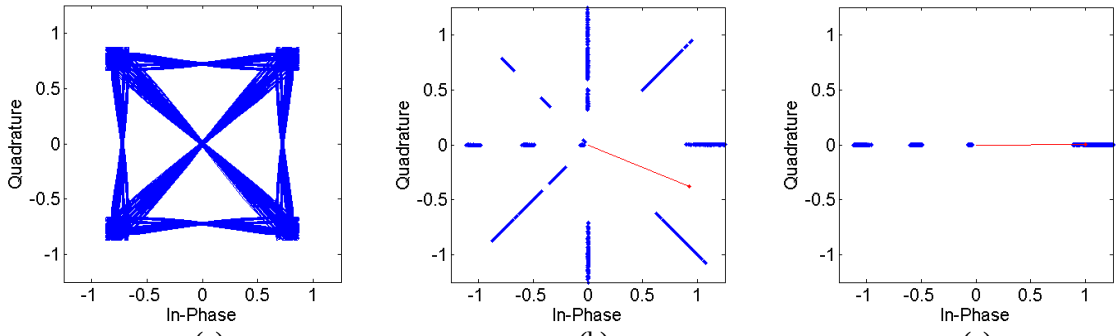
4.3 Symbol Timing Recovery

Now that the signal represents the receiver's estimate of each separate transmitted SOP, the next step taken is to recover the symbol clock. The quality of transmitter baud clock and receiver ADC sample clock is presumed to be excellent, thus stability over several thousands of symbols is presumed (and therefore timing recovery performance is not a major cause of errored symbols). The purpose of timing recovery is to first generate a stable time reference to relate the incoming samples and the sample clock to the estimated baud clock, and second to interpolate the samples to generate two samples per symbol for the equalizer. For QPSK signals (also PSK and QAM), timing recovery is performed by the NDA feed-forward "digital filter and square" method [55]. In this algorithm, the timing estimate τ (on a range of -½ to ½) is obtained by:

$$\tau = \frac{-1}{2\pi} \arg \left\{ \sum_{k=0}^{LR-1} \left| x[k] \right|^2 \exp\left(\frac{-j2\pi k}{R} \right) \right\}, \tag{4.12}$$

where L is the block length and R is the oversampling rate (i.e. samples per symbol). Note that R is generally a ratio of integers, here equaling the approximate ratio between the ADC sample clock and the baud rate.

The magnitude-squared operation (proportional to instantaneous power) is maximum at the center of symbols since the transitions in a PSK or QAM system will, with some probability, pass through regions of lower power. For example, a 180° transition between QPSK symbols will go through the origin of the I-Q plane. The spectrum of an ensemble set of $|x[n]|^2$ samples therefore has discrete tones or *spectral*



(a) (b) (c)

Fig. 4.6. 28 Gbaud QPSK transmission $x[n]$ sampled at 8x oversampling rate (a), filtered by 40 GHz 4th order super-Gaussian filter, (b) plot of timing recovery function $|x[n]|^2\exp(-j2\pi n/R)$ with R=1.6, (c) plot of timing recovery function $|x[n]|^2\exp(-j2\pi n/R)$ with R=2.

lines at $\exp(\pm j2\pi/R)$, and (4.12) downconverts the positive frequency tone to baseband and low-pass filters it. Note that these tones can be located in the second Nyquist band (i.e. $> \pi$ or $< -\pi$), and that a sampling rate of R=2 can lead to uncertainty as the tones are co-located and may not provide the optimal estimate.

The digital filter and square method can alternatively be described in the time domain on the I-Q plane. Two examples of $|x[n]|^2\exp(-j2\pi n/R)$, where $x[n]$ is a QPSK signal sampled at R=1.6 and R=2 samples per symbol, are plotted in Fig. 4.6. The original signal (Fig. 4.6a) is generated at 8 samples per symbol and decimated by 5 or 4 to achieve the target sample rate. As the timing recovery nonlinearity is averaged out, the complex vector pointing to the peak envelope is aligned (as through a weighted sum) to estimate the timing offset. For example, the bands in quadrant IV of the I-Q plane in Fig. 4.6b are all high energy (representing the eye opening) whereas the bands in quadrant II are spread among the transition types between low (180°), moderate (90°), and high energy (0°). Although the R=1.6 case (Fig. 4.6b) has a lower sample rate, its estimate is better since (unlike the R=2 case, Fig. 4.6c) the timing offset between the symbols and the sample clock continually varies. In fact the angles for the R=2 case are restricted to 0 and π because the complex exponential simplifies to $(-1)^k$. On the other hand, it is

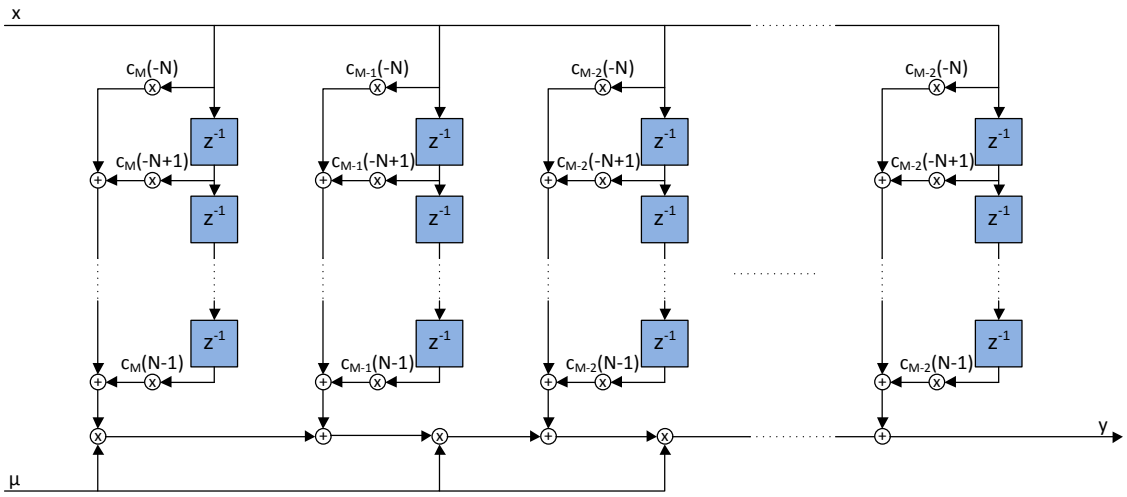


Fig. 4.7. Farrow structure for efficient implementation of polynomial or cubic Lagrange interpolator.

permissible for a signal oversampled by a factor of 2 to go on to the equalizer without interpolation, as long as symbol cycle slips are avoided by some other means. In the 100G Consortium code, it is assumed that timing recovery and interpolation must account for all symbol cycles. The timing estimate τ is forwarded to an interpolator, which generates two outputs per symbol, at τ and $\tau+T_S/2$.

Interpolation in the demodulator is accomplished by one of three methods. The first two methods are a polynomial interpolator [56], and cubic Lagrange interpolator [57], each of which use an FIR of neighboring samples to estimate of the signal at time $(m+\mu)T_S$, where m is the integer part and μ is the fractional part. The third optional method for interpolation simply upsamples the signal, runs (4.12) at the higher rate, then decimates (this method is not practical, but is helpful for code debug).

The polynomial interpolator is an FIR structure where the filter coefficients are computed to minimize quadratic error between the impulse response of the filter and an ideal interpolation filter. An implementation of the polynomial interpolator, the Farrow structure [58], is suitable for high-speed implementation (see Fig. 4.7). Performance of the polynomial interpolator depends on the number of filter taps as well as the degree of

the polynomial. The consortium's demodulator uses 6 taps on a 3rd order polynomial, which achieves <-20 dB signal degradation at BT_{Samp}=0.45.

The second option is the cubic Lagrange interpolator, based on the widely used formula from numerical mathematics. This interpolator also provides excellent performance and lends itself well to high speed implementation (Farrow structure, Fig. 4.7), though some signal degradation may occur above BT_{Samp}=0.35 [57]. This interpolator is common for 100 Gb/s systems [5, 59].

NDA timing recovery for OQPSK cannot use (4.12) since unlike QPSK the amplitude is maintained between symbols. Instead of magnitude-squared as in (4.12), the timing is estimated through a power-of-2 nonlinearity:

$$\tau = \frac{1}{4\pi}\arg\left\{\sum_{k=0}^{LR-1}\left(x[k]\right)^2 \exp\left(\frac{j2\pi k}{R}\right)\right\} - \frac{1}{4\pi}\arg\left\{\sum_{k=0}^{LR-1}\left(x[k]\right)^2 \exp\left(\frac{-j2\pi k}{R}\right)\right\} \tag{4.13}$$

The output of the $(\bullet)^2$ nonlinearity of an OQPSK signal is a antipodal signal oscillating at the symbol rate. This operation is described further in Sec. 5.3.1. Interpolation for OQPSK is identical to QPSK.

Alternative methods for NDA timing estimation are the Gardner method [60], early-late gate method [61], and Mueller-Muller timing recovery [62]. Common to all these methods is the feedback path, and hence they were not chosen.

4.4 Channel Equalization

After the signal has been interpolated to 2 samples per symbol, the signal is sent through an adaptive equalizer to undo channel effects, primarily ISI. A separate equalizer is used in this design for each polarization mode, and though it is possible to process a two-input, two-output equalizer at this stage, it has been found that there is little

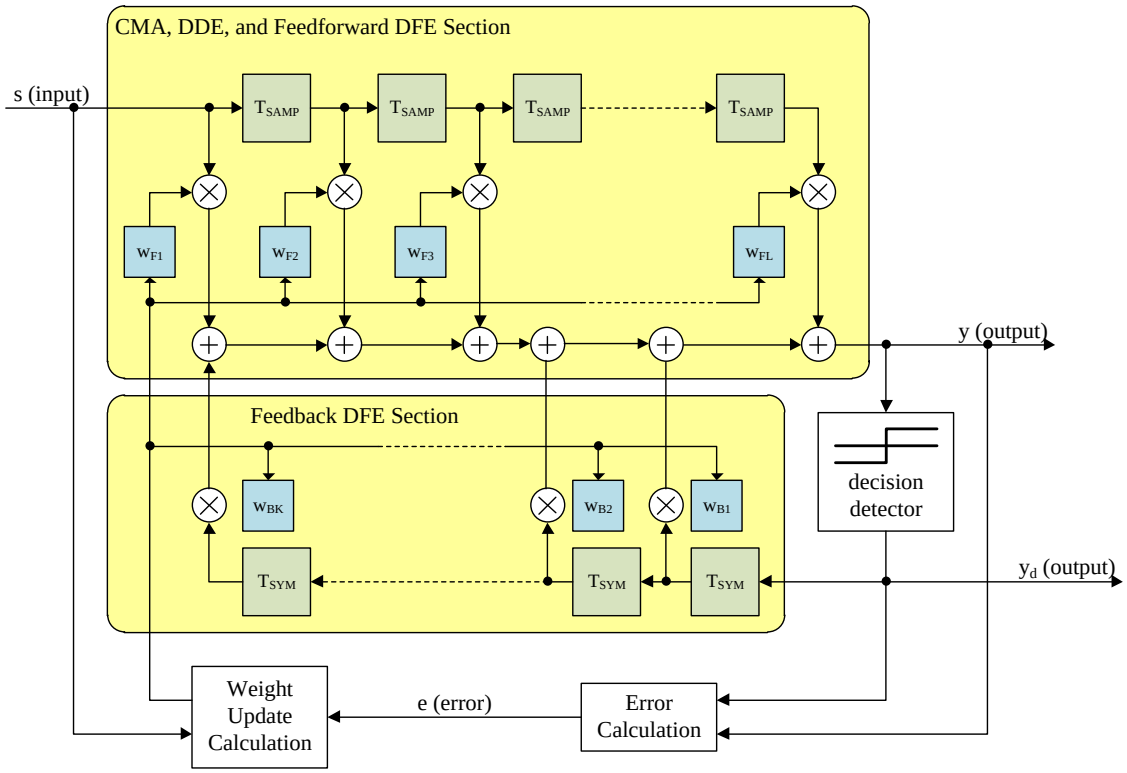


Fig. 4.8. Equalizer block diagram. Note the taps in the feedforward path are clocked at the sample ($T_S/2$ typical) rate whereas the optional DFE feedback path taps are clocked at the symbol rate (T_S).

residual cross-polarization interference after the initial polarization demultiplexer [52]. The NDA algorithm used for the equalizer is chosen among the CMA algorithm, the least mean squared (LMS) decision directed equalizer (DDE), and LMS decision feedback equalizer (DFE). All algorithms utilize the same principle structure (Fig. 4.8).

The CMA structure is used for initial acquisition, and can be also used at the onset of each block to reacquire phase offset of the new block (for burst-mode) without making decision errors. The error function for CMA is (4.8), and in the block diagram (Fig. 4.8), the decision detector is bypassed for error calculation (and the feedback DFE section is not in use). In DDE mode there is no training sequence but the decisions are presumed to be correct and are therefore used to generate the error signal $e=d-y$, where d is the complex-valued symbol decision. The DDE operates entirely out of the top section of sample-rate taps in Fig. 4.8 and, like CMA, does not use the feedback DFE section.

Performance of a DDE is generally better than CMA, under the assumption that the decisions have high probability of being correct (i.e. symbol error rate<10^{-3}).

A DFE results in the best performance by recycling prior decisions to be used in the feedback path when estimating their influence on the upcoming decisions. This DFE therefore avoids noise enhancement in the equalizer in the feedback taps. This method requires the decisions be tightly coupled to the equalizer since processing of the next symbol can only occur after the current decision is finalized, and thus is more computationally intensive than the CMA and DDE alternatives.

4.5 Carrier Phase Recovery

The final step prior to symbol detection is to estimate the phase of the carrier, a process which is complicated by the presence of phase noise on the receiver LO, ASE, and XPM. Though this module can be placed either prior to the equalizer or afterward (or be incorporated in equalizer decisions), performance in this application is better when phase is recovered after equalization. In fact, phase recovery is often the most sensitive aspect of the coherent optical demodulator since XPM, ASE, and laser linewidth combine to strongly influence the phase.

In general, the k^{th} symbol input to the phase estimator is

$$r[k]=x[k]\exp(j\theta[k])+n[k], \tag{4.14}$$

where $x[k]$ is the modulated signal, $\theta[k]$ is the phase noise term, and $n[k]$ is a separate additive noise term. For QPSK, the signal can be defined over an alphabet $\boldsymbol{\alpha}=(\pm 1, \pm 3)$:

$$x[k] = \frac{\sqrt{E}}{T_s}\exp\left(\frac{j\alpha\pi}{4}\right). \tag{4.15}$$

The preferred method of phase recovery is the power-of-4 nonlinearity, which strips the signal of its modulation:

$$r[k]^4 = x[k]^4 \exp(j4\theta[k]) + n[k]^4 = -1 \cdot \exp(j4\theta[k]) + n[k]^4 \quad (4.16)$$

Averaging this out over a window of N_{CR} symbols to estimate phase is known as the Viterbi & Viterbi method [63]:

$$\theta = \frac{1}{4} \arg\left\{ \sum_{k=-L/2}^{L/2} \left(x[k] \right)^4 \right\}, \quad (4.17)$$

where L is the block length. The Viterbi & Viterbi algorithm is adequate, though the potential exists for much improvement in this area.

100G Consortium partner Optametra uses a Wiener filter for its phase recovery, given the analysis of the optimal for tracking laser phase noise [64]. Laser phase noise, specified by linewidth, is a Wiener process, where the phase of the next symbol takes a random incremental trajectory, zero mean and independent of previous steps (also called "random walk"). The filter matching this behavior has an impulse response with a peak value on the symbol of interest and an exponential decay on either side of the peak. The rate of decay is proportional to linewidth. Unfortunately, this analysis neglects the influence of ASE and XPM, both of which severely degrade performance of this method.

A very detailed analysis was conducted by another 100G Consortium partner, OFS, to generate a minimum mean-squared error (MMSE) filter accounting for ASE in addition to the laser phase noise. Without divulging the intellectual property of the analysis, the resulting filter is a hybrid between the flat averaging window of the Viterbi & Viterbi method and the Wiener filter method. At high OSNR, the filter is similar to the Wiener filter, whereas at low OSNR there is need to average over several symbols to mitigate the impact of ASE noise on the estimate. This method continually computes an

estimate of the autocorrelation of the 4^{th}-power of the phase, but requires an accurate OSNR estimate to generate the MMSE filter coefficients. In the experimental setup, this method has very similar performance to the Viterbi & Viterbi method at low OSNR, and is slightly worst in an XPM environment since the OSNR belies the true variation in the signal (i.e. signal power increases above noise floor, but is also the source of additional nonlinear phase noise).

Though all methods are available in the demodulator code, Viterbi & Viterbi is preferable due to its relative simplicity, as it does not need to estimate the autocorrelation of the phase estimate. Another method for phase estimation is to use a PLL, updated by the phase difference between the decision value and the symbol value. Since the QPSK format can take advantage of a feed-forward implantation like those discussed above, the feedback feature of the PLL makes it less desirable. However, the PLL method must be given strong consideration for higher order QAM signals and other more complex formats.

A second-order PLL implementation tracks not only phase but also frequency. Though frequency tracking is not strictly necessary for the Viterbi & Viterbi (or other NDA feed-forward symmetric filter types), it nevertheless improves performance when it is provided. In keeping with the block processing methodology, frequency offsets are identified by using an FFT to detect a peak tone in DFT$\{r[k]^4\}$, and then eliminated prior to phase recovery. Also within the phase recovery loop is an angle-unwrap function, which minimizes phase jumps when the estimate crosses a 90° boundary.

Like timing estimation, the OQPSK phase estimate requires a different algorithm than QPSK. Fortunately, the formula for phase estimation is very similar to timing estimation (4.13), so it is possible for them to share processing:

$$\theta = \frac{1}{4}\arg\left\{\sum_{k=0}^{LR-1}\left(x[k]\right)^2 \exp\left(\frac{j2\pi k}{R}\right)\right\} + \frac{1}{4}\arg\left\{\sum_{k=0}^{LR-1}\left(x[k]\right)^2 \exp\left(\frac{-j2\pi k}{R}\right)\right\}. \quad (4.18)$$

Note that the timing estimate was a difference term between the pair of nonlinearities, whereas the phase estimate is a summation. Frequency offset can be readily detected from an offset in the spectral lines in the positive and negative frequency space.

4.6 Pattern Generation, Matching, and Differential Decoding

After phase recovery, a practical demodulator makes a hard decision on the symbol and passes the result on to the FEC decoder (separate from the demodulator here). In the 100G Consortium's demodulator code, the signal is instead processed by a PRBS matching unit to report the BER. The data source at the transmitter may be the bit error rate tester (BERT) in the lab, or just a PRBS generator component in the case of simulation. A typical generator polynomial is $x^{15}+x^{14}+1$ for a run length of $2^{15}-1$. A block diagram of this generator is depicted in Fig. 4.9.

Matching to the pattern is attempted by shifting in bits to the generator structure (Fig. 4.9) and monitoring the output. Once a string of matches has been found, the match unit is then allowed to run freely and count the number of mismatches between its output and the received bit pattern. Depending upon the phase state of the demodulator, the

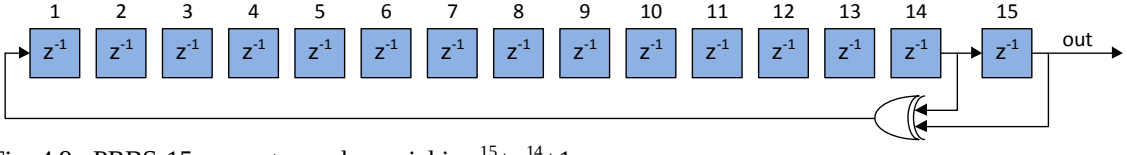


Fig. 4.9. PRBS-15 generator, polynomial is $x^{15}+x^{14}+1$

received sequence may need to be inverted to match the intended pattern. In a practical system, framing overhead is added to properly align the data pattern to avoid such an inversion.

The demodulator code can optionally perform differential decoding, which avoids the issue of 90° phase slips that result in long bursts of errors. If the symbols are differentially encoded, a 90° phase hit will result in a single bit error. On the other hand, any symbol error caused by ASE or XPM will result in two bit errors; one error for the incorrect decision, and a second error on the following decision since the correct phase difference also depends on the errored symbol. Regardless of the setting for differential detection on the receiver the transmitted data pattern is PRBS-15 in this system; it is not differentially encoded prior to transmission. To accommodate differential decoding, the received signal is searched for the first pattern match on the raw decisions. The pattern is then generated starting from the first match, the length of which matches the demodulator data output length. Finally, both signals (demodulator output and pattern sequence) are differentially decoded as though they were properly encoded, and symbol errors are counted among the decoder outputs. A block diagram illustrating this operation is shown in Fig. 4.10.

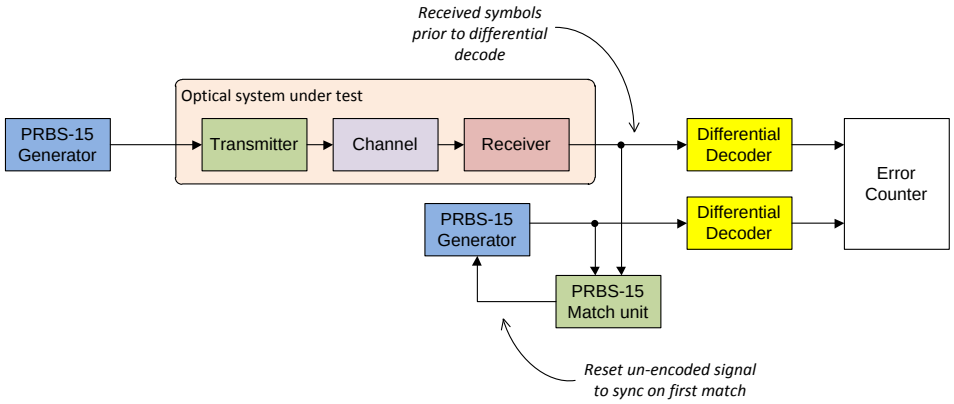


Fig. 4.10. Pattern matching block diagram for simulating differentially decoding, where transmitted data is uncoded.

CHAPTER 5

CPM OVER FIBER

5.1 CPM Parameter Selection

As described in Ch. 3, several parameters combine to define a CPM signal. Together the parameters determine the spectral efficiency, minimum distance, and implementation complexity of the system. The fast sequential method for computing minimum distance is accurate and computationally efficient for unfiltered CPM signals, though it does not account for the effects of a channel filter [37]. In a fiber-optic application, we are concerned with optimizing performance over a bandlimited channel, where the ROADM channel filters are independent from the modulator.

5.1.1 Modifications to fast sequential algorithm

The fast sequential algorithm requires adaptation to accommodate filtering. Unfortunately, minimum distance computation of a filtered signal cannot be simplified by a phase difference tree, since the high-frequency energy that is rejected by the filter influences both phase and amplitude of the received signal. In other words, the symbol sequences that would result in overlapped paths on the phase difference tree in an unfiltered environment are no longer guaranteed to overlap, nor is the envelope constant, and therefore the simplification from (3.9) to (3.10) no longer holds. However, it is still possible to utilize both the upper bound and process of elimination from the original sequential algorithm.

The sequential algorithm has been modified to account for filtering effects, as described in the flow chart in Fig. 5.1. The algorithm starts by calculating the upper bound limit at the initial modulation index h_{min}, using (3.11) as for the unfiltered signal, noting that it is impossible to achieve a higher minimum distance for a non-degenerate filtered CPM signal than its unfiltered counterpart. After the upper bound is calculated, the permutations of CPM signals begin with N=1, analyzing the minimum distance between phase trajectories over a single symbol interval. In the filtered algorithm, the CPM signal pattern under analysis is preceded by symbols with the maximum phase trajectory (M-1) for a duration of (N+L+1) symbol periods, and again following the pattern for another (N+L+1) symbol periods. By doing so, the filtered signal will achieve maximum frequency offset, and will suffer maximum attenuation by the ROADM filter at the onset and conclusion of the analyzed patterns.

Continuing through the flow chart, the minimum distance between different symbol patterns is computed directly from the filter output using (4). No generalization can be made to specify an incremental minimum distance for the filtered signal as is made in (7) for unfiltered signals. As the algorithm progresses along N (from 1 to N_{max}), the surviving symbol sequences (of length N-1) are regenerated as entire patterns, as the integral for Eq. 4 spans N symbols. Again, after all possible sequences are attempted any symbol sequences that exceed the upper bound are discarded prior to incrementing N. The algorithm terminates once N reaches N_{max}.

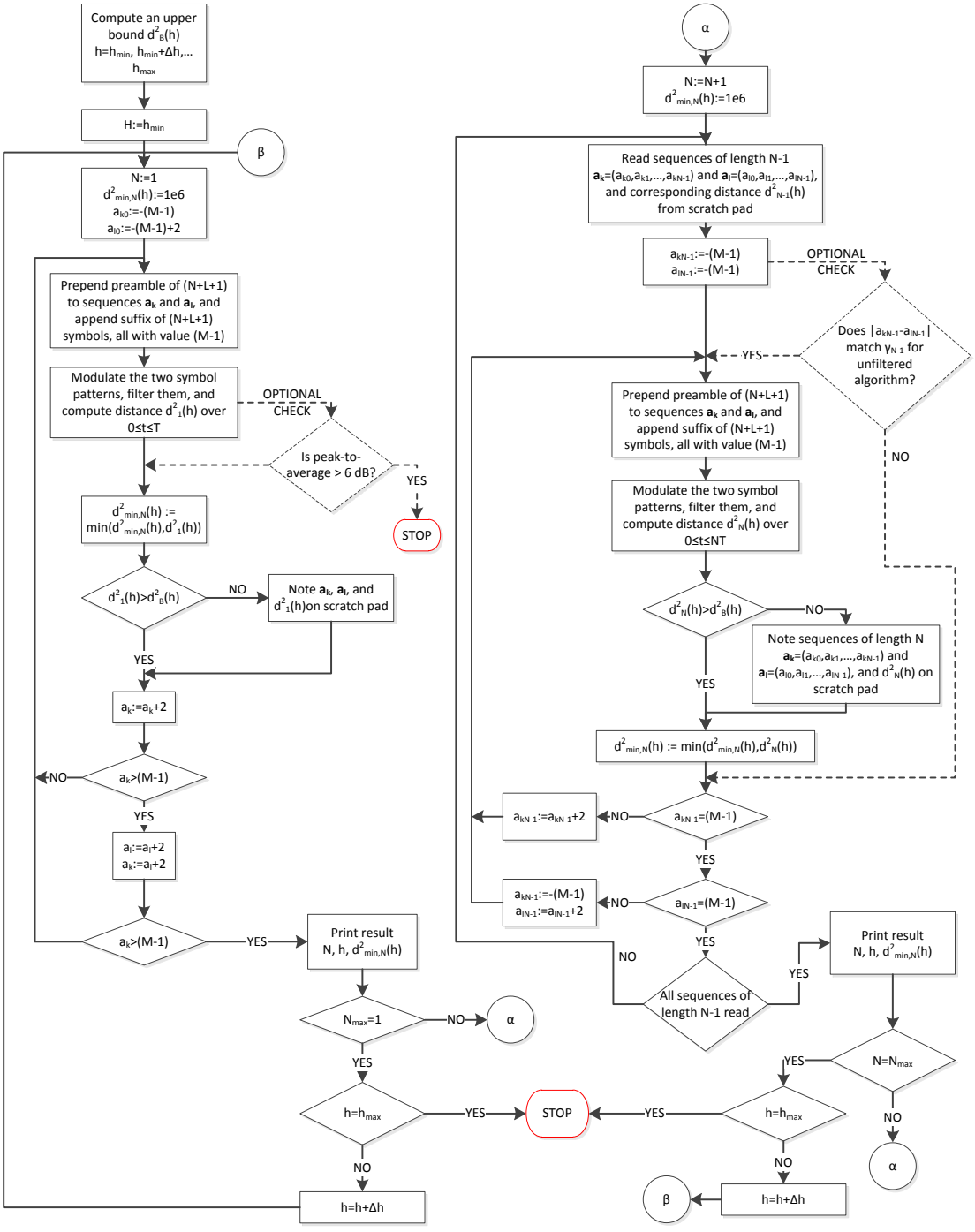


Fig. 5.1. Flowchart of modified sequential algorithm for computing minimum distance of arbitrary filtered CPM signals over specified ranges of *h* and *N*

From empirical data, there are two classes of behavior for filtered signals to achieve the minimum distance as they proceed through this modified algorithm. The first is when the attenuation of the filter is so great for those sequences with maximal phase

trajectory that minor differences in the sequence result in little difference in the filter output. This class of behavior is indicative that the scheme under study no longer has low intensity fluctuation (and correspondingly higher PAPR). The second behavior for filtered signals is that the symbol sequence that achieves minimum distance corresponds directly to the phase difference with the minimum distance sequence of the unfiltered signal. This fact can be leveraged to dramatically reduce the number of computations in the modified algorithm to achieve the same result by first calculating minimum distance for the scheme in an unfiltered environment, then examining the surviving paths under filtering.

5.1.2 Minimum distance through ROADMs

To quantify the effect on minimum distance due to the presence of ROADM filters, computations were performed with the modified fast sequential algorithm. The filter model has a super-Gaussian transfer function (2.27), which is characteristic of many ROADM filters. From [36] it has been shown that $M>4$ provides little or no benefit for CPM schemes with SE>1 b/s/Hz, and the computations are therefore limited to binary and quaternary schemes. Binary schemes are significantly less complex to generate and receive, though they lack the required SE for 100 Gb/s transmission over a 50 GHz grid. Calculations were performed over full response and partial response schemes using the REC and RC pulse shapes. The number of ROADMs (40 GHz passband, order 3.5) in the simulation varied over the set of 0 (unfiltered), 1, 3, and 8.

Minimum distance is plotted versus h in for select CPM schemes in Figs. 5.2 and 5.3. The binary full response schemes, shown in Fig. 5.2, indicate that filtering has little effect through the first local maxima, which occurs near h=0.7 for 1REC and h=0.6 for

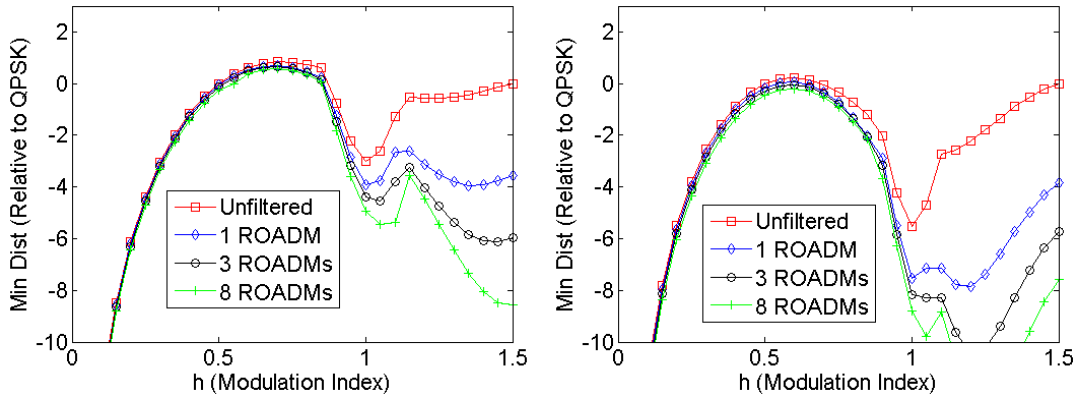


Fig. 5.2. Normalized minimum Euclidean distance versus modulation index for binary full response CPM schemes (a) 1REC, and (b) 1RC, observation interval N=4 symbols. Filtering through 0, 1, 3, or 8 super-Gaussian filters with 40 GHz passband, order 3.5.

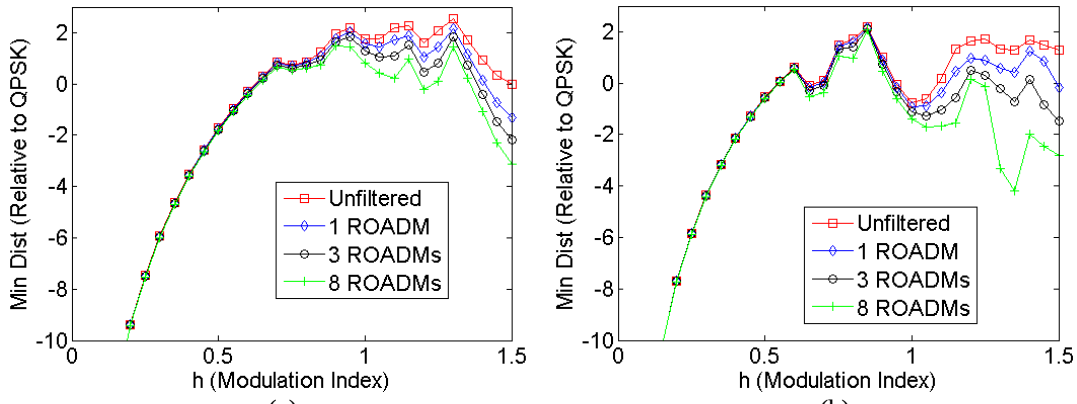
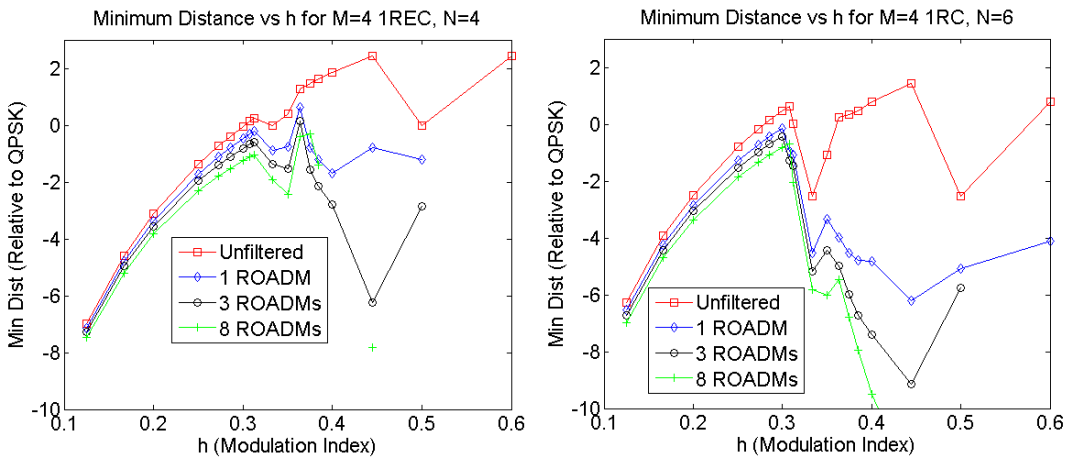


Fig. 5.3. Normalized minimum Euclidean distance versus modulation index for binary partial response CPM schemes (a) 3REC, and (b) 3RC, observation interval N=6 symbols. Filtering through 0, 1, 3, or 8 super-Gaussian filters with 40 GHz passband, order 3.5.

1RC. As h increases above 1.0, the cascade of filters degrades the available minimum distance as the signal is attenuated during instants of high frequency deviation. Naturally, the conclusion for a bandlimited channel is to avoid higher modulation indexes, which lead to lower SE without, in the binary full response case, enhancing minimum distance.

Binary partial response schemes (Fig. 5.3) benefit from the pre-filtering of the symbols, which does not decrease the maximum frequency excursion, though it reduces the likelihood of encountering it. The 3REC and 3RC schemes are robust to filtering through the range $h \leq 1$, with optimal performance near h=0.9 for 3REC and h=0.85 for



(a) (b)

Fig. 5.4. Normalized minimum Euclidean distance versus modulation index for quaternary full response CPM schemes (a) 1REC, N=4, and (b) 1RC, N=6. Filtering through 0, 1, 3, or 8 super-Gaussian filters with 40 GHz passband, order 3.5.

3RC. Unlike full response, the raised-cosine partial response scheme achieved better performance than the rectangular pulse scheme.

Moving beyond binary to the quaternary schemes, the maximal frequency offset occurs at the symbol value α=±3, triple that of corresponding binary schemes. Thus, in Fig. 5.4, it is observed that the full response quaternary schemes peak in the vicinity of h=0.3 to h=0.375, and quickly degrade above that region. The peak frequency offset of 1REC (CPFSK) is lower, which allows for higher h than the 1RC scheme. Note that quaternary CPFSK schemes with modulation indices above 0.3 achieve equivalent or higher minimum distance in an unfiltered system compared with QPSK (see Fig. 3.4c).

Unlike the 1REC scheme, minimum distance of the 1RC pulse shape fluctuates between underperforming and outperforming that of QPSK in an unfiltered application (see Fig. 3.4d). Additionally, SE is lower than 1REC, so the filtering impairs the signal at lower modulation indexes as shown in Fig. C. As a result, the scheme does not quite reach the same level of performance as 1REC in the region of interest between h=0.3 and

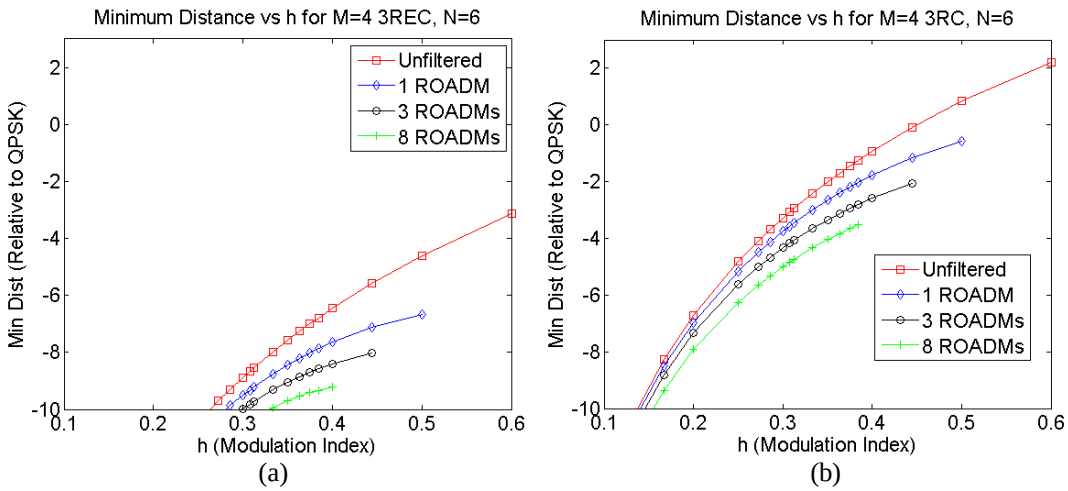


Fig. 5.5. Normalized minimum Euclidean distance versus modulation index for quaternary partial response CPM schemes, (a) 3REC and (b) 3RC, with observation window N=6 symbols. Filtering through 0, 1, 3, or 8 super-Gaussian filters with 40 GHz passband, order 3.5.

h=0.375.

Likewise, the partial response 3REC and 3RC quaternary schemes require higher modulation index than full response schemes to outperform QPSK. Indeed, 3RC can achieve over 5 dB gain versus QPSK in an unfiltered case (see Fig. 3.5) but this occurs at h=1.3, which operates at a SE of 0.48 b/s/Hz. In other words, a 28 Gbaud quaternary system would require a passband of roughly 116 GHz to maximize performance of the 3RC scheme. Coupled with the fact that implementation complexity increases dramatically with increase in L, partial response quaternary systems are not especially compelling for fiber links in the 100 Gb/s application. If in the future a 40 Gb/s application requires 5 dB better performance on a 50 GHz grid, for example, a 3RC quaternary CPM system may be worth consideration.

5.2 Optical CPM Signal Generation

With the guidance from the parameter selection in Sec. 5.1, the objective is to design a system capable of generating and receiving a CPFSK signal with modulation index in the range of roughly 0.25 to 0.375. On the transmitter side, an external

modulator architecture is preferred for both temporal and amplitude resolution. As mentioned in Sec. 2.3, two available devices for such an architecture are the phase modulator (PM) and the nested MZM quadrature modulator (QM). Using these components, there are two proposed architectures for CPM signal generation, including a novel hybrid PM-QM approach.

The first option for transmitter architecture is to leverage the separate I and Q controls available in the QM, and drive them with a set of DACs that operate at a sample rate (in GHz) higher than the passband of the channel filter. In this application, assuming a 40 GHz channel filter, a sample rate of 50 GHz would provide sufficient alias rejection to support the generation of an arbitrary complex waveform. However, the main issue with such an approach is the availability of DACs at the sample rate of interest. Though such devices exist, their resolution is currently limited to 4 or 5 effective number of bits (ENoB). This factor limits the available performance, unless the sample rate and modulation scheme are chosen under this constraint. It is therefore recommended for such an architecture to operate with a sample rate of twice the baud rate, and with modulation $h=m/p$ (m and p are positive integers) chosen to minimize the total number of phase states. Recall from (3.7) and (3.8) there are p states for even m and $2p$ states for odd m. The total number of signal states at the transmitter doubles for full response signaling when issuing samples at twice the baud rate.

The QM architecture is illustrated in Fig. 5.6, with example control voltage scenarios. The first example shows $h={}^1/_3$, which is the simplest choice of modulation index within the range of interest. In this scheme there are six possible phase states, though only three are reachable per symbol (the ±3 symbol values converge on the same

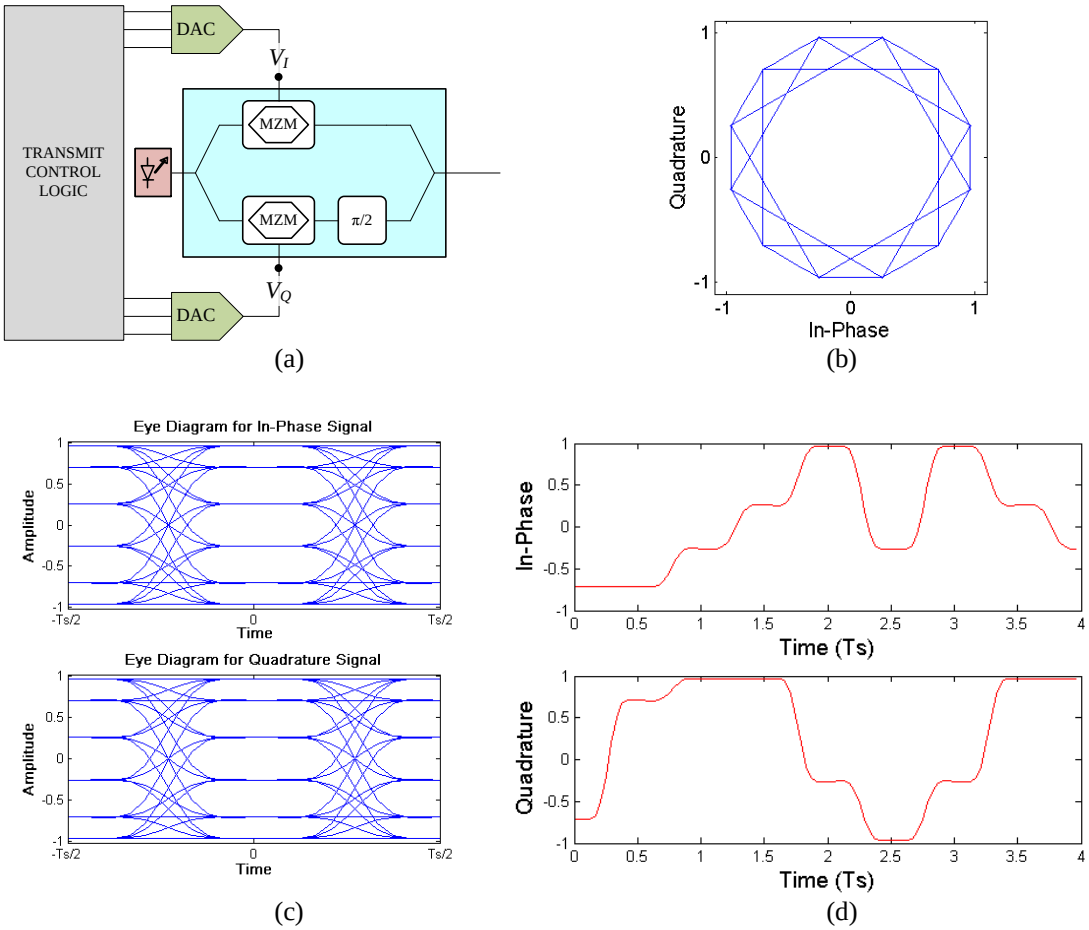
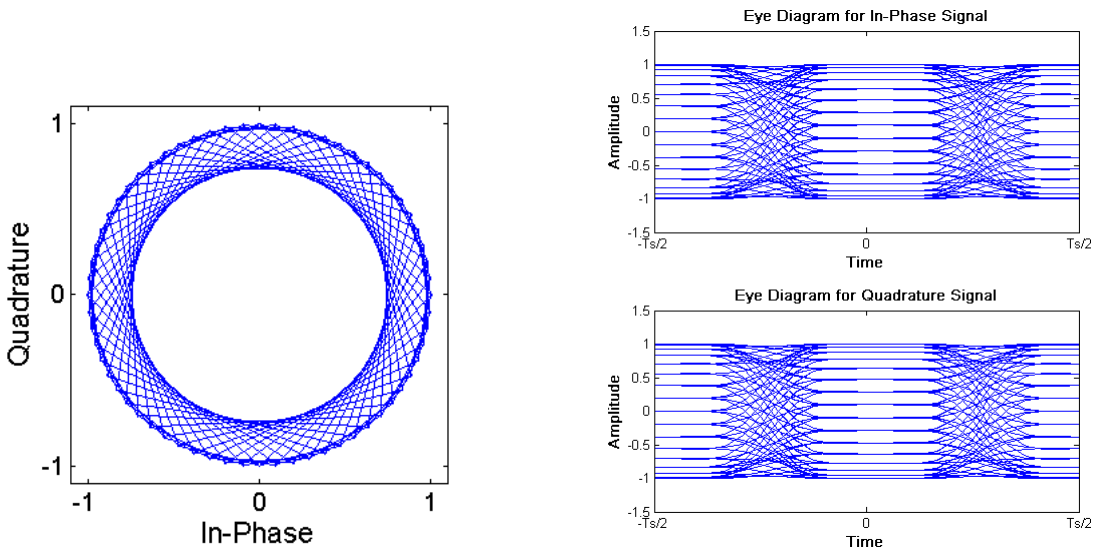


Fig. 5.6. QM-based transmitter example for a quaternary h=1/3 format with 2x oversampling: (a) modulator block diagram, including digital controls, DACs, QM, and laser source, (b) generated constellation, (c) eye diagrams and (d) example waveforms for control voltages.

final phase state as they take trajectories of length ±π). Including the half-symbol phase states, there are only 12 discrete points on the I-Q plane that must be reached by the QM. In an unfiltered application, this CPFSK scheme has identical minimum distance as QPSK, though it has lower minimum distance than schemes with *lower* modulation index in the range $0.3<h<{}^{1}/_{3}$, due to the lack of diverse phase states. So there is a tradeoff between simplicity of implementation (affecting both transmitter and receiver) and minimum distance.

A second scenario for the QM in Fig. 5.7 utilizes $h={}^{5}/_{16}$ (=0.3125), which has higher minimum distance due to the diversity of states and due to its higher SE (reducing



(a) (b)

Fig. 5.7. QM-based transmitter example for a quaternary h=5/16 format with 2x oversampling: (a) generated constellation, and (b) eye diagrams for control voltages.

the impact of the channel filter). Though other ratios with higher minimum distance are available (i.e. $h = {}^{8}/_{25} = 0.32$), this choice has the benefit of a power-of-two number of phase states on the symbol boundary (32) and half-symbol boundary (64). Due to the large number of states, implementation complexity (for both the transmitter and receiver) is still a major concern, hence the choice of this scenario offers a different balance of performance and complexity than $h={}^{1}/_{3}$.

As depicted in Figs. 5.6 and 5.7, the DAC-driven QM approach does not actually produce a CPM signal, since the transitions between phase states are not confined to the unit circle. Though this is a limitation, the fact is that the presence of the channel filter cuts off the high-frequency energy components that are necessary to maintain constant intensity. At the modulation indices of interest, the percentage of signal power in the sidelobes that is cut off is significant (roughly 5 to 10 percent). There is, however, an alternative architecture that allows the transmitter to maintain constant intensity, while maintaining low complexity.

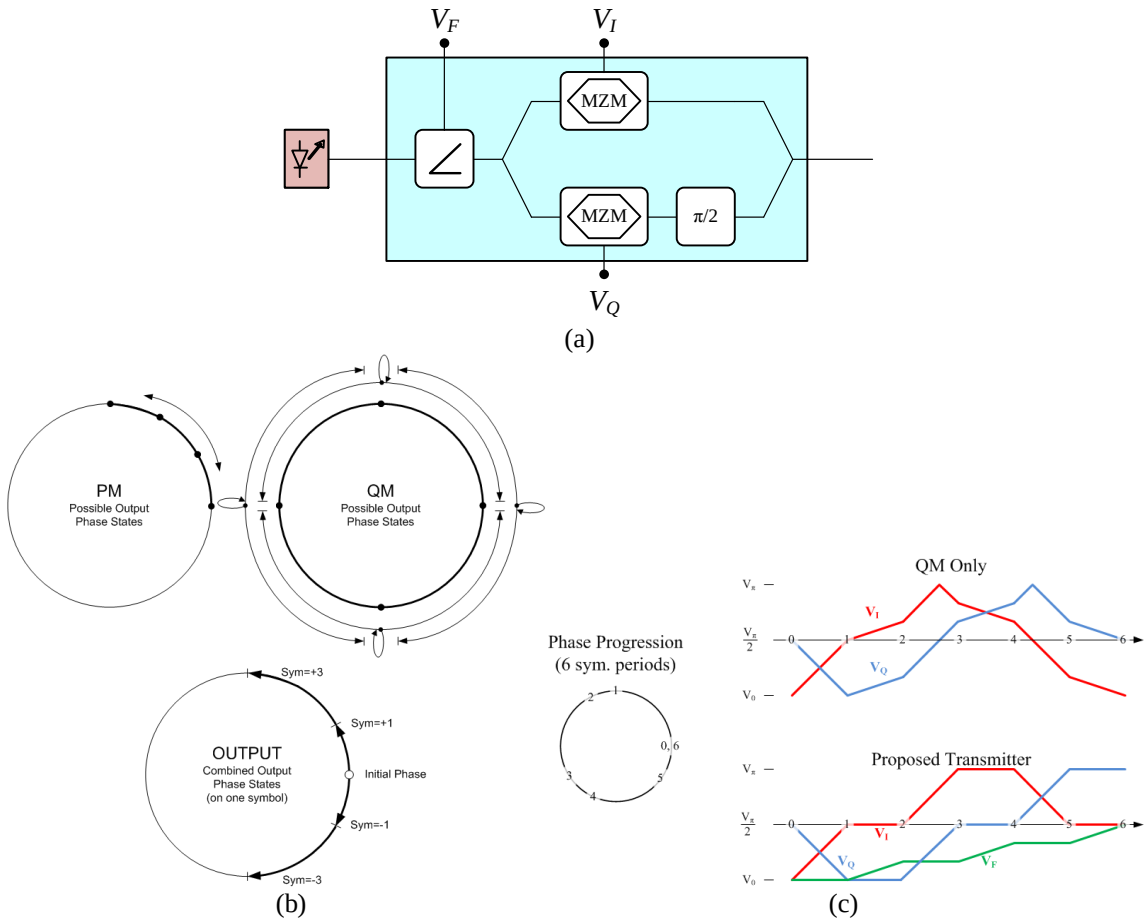


Fig. 5.8. PM-QM transmitter example for a quaternary h=1/6 format: (a) modulator block diagram, including 4-level phase control and 3-level I and Q controls, (b) diagram of phase trajectory development from coordinated PM and QM transitions, and (c) example pattern for development of phase states, contrasting QM-only approach (maintaining constant amplitude) versus the proposed transmitter.

The PM-QM architecture (Fig. 5.8 [5.3]) consists of a single PM and one or more QM structures, all synchronized with each other, and all in cascade. In this structure the role of the PM is to provide fine-tune phase shift capability within its finite tuning range. The QMs are driven to achieve the coarse-tune phase shifts, allowing an unbounded phase progression that is required by definition of a CPM signal from (3.1) and (3.2). The driving voltages of the QMs comprise three discrete levels on each of the I and Q lines and, when inter-dependently modulated at the symbol rate, enable phase transitions of 0 and ±π/2 around the unit circle. Together with the sinusoidal transfer function of the

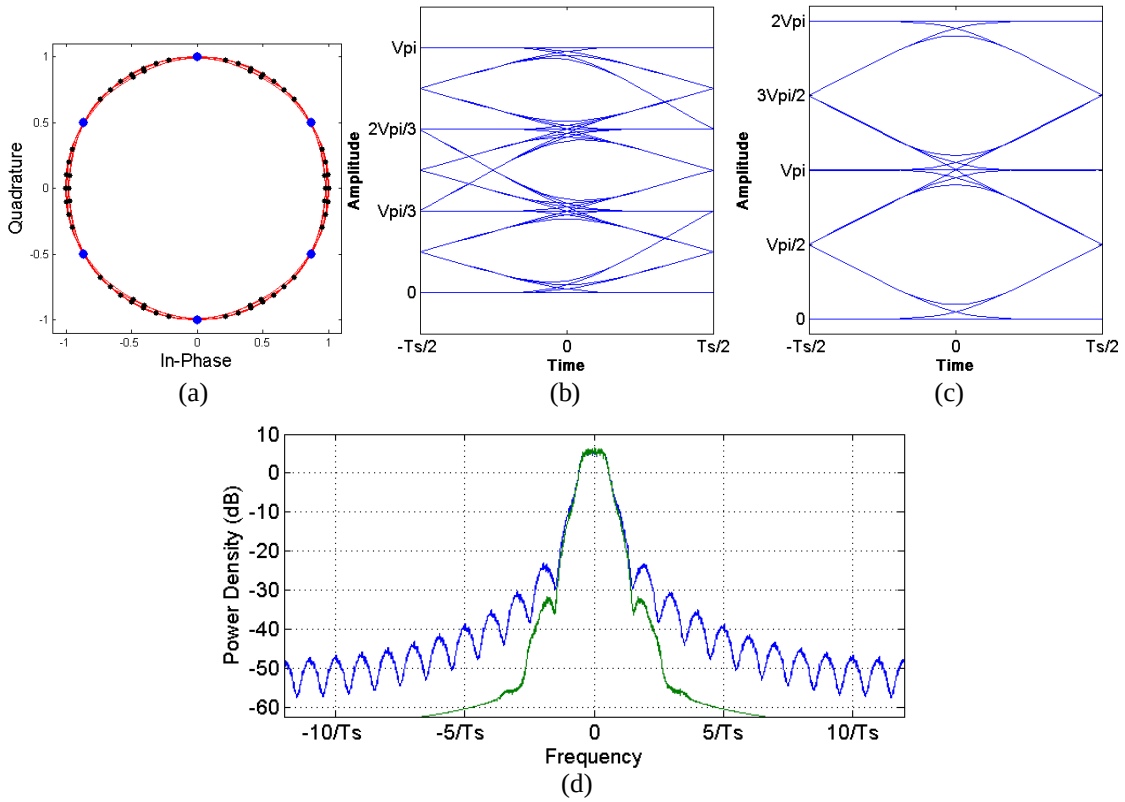


Fig. 5.9. PM-QM signal generation example, M=4, h=1/3, 1REC, with low-pass filtered signals on PM and QM control lines: (a) constellation of transmitted signal with black dots on symbol boundaries and blue dots at symbol centers, (b) eye diagram of filtered PM control signal, (c) eye diagram of filtered QM control signal, (d) spectra of the filtered (green) vs. unfiltered (blue) signal.

MZM (2.26), constant intensity is maintained throughout the QM symbol progression with a linear ramp between control voltage levels.

Drive voltages for a quaternary CPFSK h=1/6 example are inlaid to Fig. 5.8. A separate h=1/3 example are shown in Fig. 5.9, where the unit is required to have 2 QMs and 1 PM. This architecture generally requires $\lceil 6h \rceil$ QMs in a quaternary full response system, since each one has a maximum range of $\pm\pi/2$ per symbol, and the maximal phase offset between adjacent symbols is $\pm 3\pi h$. The signal does not lose much of its envelope constancy even though the QM drive voltages (e.g. 5.9b) are filtered instead of the ideal triangular ramps. The phase of the output signal at the symbol boundaries is smeared due to the filtering of QM and PM drive signals, though this ISI is common for any bandpass-

filtered CPM signal type (this assertion is supported by the spectra plot in Fig. 5.9d). Similar to previous work on optical multi-level MSK [28], there is a range of acceptable filtering or non-ideal behavior on the drive voltage without degradation of modulated signal integrity. Note the total number of control voltages for h=1/3 (5 for that case) is more than double that of the QM-only architecture (requires only 2) from Fig. 5.6.

5.3 Optical CPM Reception Techniques

Chapter 4 presented an overview of the demodulator functions performed for coherent QPSK reception. Though CPM reception is considerably different than QPSK at the symbol detector, many of the same functions are performed ahead of it. For example, the chosen strategies for CD compensation (Sec. 4.1) and polarization demultiplexing (Sec. 4.2) apply directly to CPM. Indeed, the asynchronous polarization demultiplexer operation for CPM is well matched since the signal passes through the channel with nearly constant amplitude (depending on the amount of attenuation). The differences for CPM lie primarily in synchronization, equalization, and symbol detection.

5.3.1 Synchronization

Timing and carrier recovery in CPM systems may be performed in a variety of ways, but again feed-forward NDA methods are highly favorable. Though they differ from QPSK, two methods for CPM synchronization were presented in Ch. 4 as they pertain to OQPSK; the estimates from (4.13) and (4.18) are for binary, h=0.5, full response CPM systems such as MSK. The methods apply to OQPSK since these formats all alternate between four phase states at 90° offsets in the I-Q plane, with two possible symbols at a time. To generalize the methods, consider a received signal:

$$r(t;\boldsymbol{\alpha}) = s(t;\boldsymbol{\alpha}) + n(t)\,, \tag{5.1}$$

where $n(t)$ is additive noise and $s(t;\boldsymbol{\alpha})$ is the modulated signal from (3.1), with $h=m/p$ (m and p are integers). Normalizing signal power and neglecting the noise term, the received signal with carrier offset f_c and modulated phase $\varphi(t,\alpha)$ from (3.2) is:

$$r(t;\boldsymbol{\alpha})^p = \frac{(2E_S)^{p/2}}{T_S{}^{p/2}} \cos{}^p\left(2\pi f_c(t-\tau) + \varphi(t;\boldsymbol{\alpha}) + \theta\right), \tag{5.2}$$

where τ and θ are the time-varying symbol timing and carrier phase offsets between transmitter and receiver. The power-of-p function (5.3) expands to

$$r(t;\boldsymbol{\alpha})^p = \frac{(2E_S)^{p/2}}{T_S{}^{p/2}} \left\{ \frac{2}{2^p} \sum_{k=0}^{(p-1)/2} \binom{p}{k} \cos\left([\,p-2k\,][\,2\pi f_c(t-\tau) + \varphi(t;\boldsymbol{\alpha}) + \theta\,]\right) \right\} \text{ for odd } p \text{ and} \tag{5.3}$$

$$r(t;\boldsymbol{\alpha})^p = \frac{(2E_S)^{p/2}}{T_S{}^{p/2}} \left\{ \frac{1}{2^p}\binom{p}{\frac{p}{2}} + \frac{2}{2^p} \sum_{k=0}^{p/2-1} \binom{p}{k} \cos\left([\,p-2k\,][\,2\pi f_c(t-\tau) + \varphi(t;\boldsymbol{\alpha}) + \theta\,]\right) \right\} \text{ for even } p. \tag{5.4}$$

When the signal (5.3) or (5.4) is narrowband filtered around the frequency pf_c, it simplifies to

$$r(t;\boldsymbol{\alpha})^p = \frac{(2E_S)^{p/2}}{T_S{}^{p/2}\,2^{p-1}} \cos\left[2\pi p f_c(t-\tau) + p\,\varphi(t;\boldsymbol{\alpha}) + p\,\theta\right]. \tag{5.5}$$

This is a CPM signal with carrier frequency pf_c, whose phase is modulated with the integer modulation index m:

$$p\,\varphi(t;\boldsymbol{\alpha}) = p\,2\pi\,\frac{m}{p} \sum_k \alpha_k q(t-kT_S)\,. \tag{5.6}$$

Spectral analysis of CPM signals [36] reveals that a CPM signal with integer modulation index has discrete $1/T_S$-spaced frequency components at offsets $\{0, \pm 1/T_S, \pm 3/T_S, \ldots\}$ for even h, and $\{\pm 1/(2T_S), \pm 3/(2T_S), \ldots\}$ for odd h. In the special case of CPFSK, there are only M tones in the signal from (5.5) since each symbol value produces

a distinct tone. These tones are located at $[pf_c+\alpha m/(2T_S)]$, where $\alpha=\{\pm1, \pm3, \ldots \pm(M\text{-}1)\}$, the lower two of which can be isolated by LPF:

$$r_{LPF}(t;\boldsymbol{\alpha})^p = \beta_1 \exp\left\{ j2\pi(t-\tau)\left(pf_c + \frac{m}{2T_S} \right) + jp\theta \right\} + \beta_2 \exp\left\{ j2\pi(t-\tau)\left(pf_c - \frac{m}{2T_S} \right) + jp\theta \right\}. \quad (5.7)$$

Here the factors β_1 and β_2 are equal when the symbols ±1 are equiprobable and filtering attenuates the tones equally (which may not be the case for a large receiver LO offset).

Timing and phase estimates are established by frequency shifting both tones to baseband (i.e. $\exp\{\pm j2\pi t[m/(2T_S)]\}$), then lowpass filter:

$$z_{LSB}(t) = \text{LPF}\left\{ r_{LPF}(t;\boldsymbol{\alpha})^p \exp\left(\frac{j2\pi tm}{2T_S} \right) \right\} = \beta_2 \exp\left\{ j2\pi(t-\tau)pf_c + \frac{j2\pi\tau m}{2T_S} + jp\theta \right\}, \text{ and} \quad (5.8)$$

$$z_{USB}(t) = \text{LPF}\left\{ r_{LPF}(t;\boldsymbol{\alpha})^p \exp\left(-\frac{j2\pi tm}{2T_S} \right) \right\} = \beta_1 \exp\left\{ j2\pi(t-\tau)pf_c - \frac{j2\pi\tau m}{2T_S} + jp\theta \right\}. \quad (5.9)$$

The NDA estimates using the two signals $z_{LSB}(t)$ and $z_{USB}(t)$ are:

$$\tau = \frac{T_S}{\pi m}\arg\{z_{LSB}(t)\} - \frac{T_S}{\pi m}\arg\{z_{USB}(t)\}, \text{ and} \quad (5.10)$$

$$\theta = \frac{1}{2p}\arg\{z_{LSB}(t)\} + \frac{1}{2p}\arg\{z_{USB}(t)\} - 2\pi f_c(t-\tau). \quad (5.11)$$

Note that the timing estimate (5.10) is independent of carrier phase, and that the phase estimate (5.11) is independent of timing if τ is slowly varying or if $f_c \approx 0$. Thus, timing and carrier recovery can be performed jointly if desired.

Recall from Ch. 4 that the sampling rate at the ADCs is R, a ratio of two integers. A frequency shift for (5.7) to baseband of $\exp(\pm j\pi tm/T_S)$ at a sampling rate of R/T_S corresponds to a shift of $\pm\omega m/R$ in the discrete frequency space. In the discrete time domain, for received signal $x[n]$ with sample rate R/T_S, the estimates are:

$$\tau = \frac{1}{2\pi m}\arg\left\{\sum_{k=0}^{LR-1}\left(x[k]\right)^{p}\exp\left(\frac{j2\pi km}{R}\right)\right\} - \frac{1}{2\pi m}\arg\left\{\sum_{k=0}^{LR-1}\left(x[k]\right)^{p}\exp\left(\frac{-j2\pi km}{R}\right)\right\}, \text{ and} \quad (5.12)$$

$$\theta = \frac{1}{2m}\arg\left\{\sum_{k=0}^{LR-1}\left(x[k]\right)^{p}\exp\left(\frac{j2\pi km}{R}\right)\right\} + \frac{1}{2m}\arg\left\{\sum_{k=0}^{LR-1}\left(x[k]\right)^{p}\exp\left(\frac{-j2\pi km}{R}\right)\right\}, \quad (5.13)$$

where L is the block length (not necessarily equal for timing and phase recovery modules), and the LPF is achieved by the unweighted average over each block. Again, the equations for OQPSK synchronization (4.13) and (4.18) correspond to those used for MSK, which is a CPFSK binary scheme with h=0.5 (m=1, p=2).

One alternative method exists for timing recovery of non-binary CPM signals, which is to filter the complex baseband signal with an LPF whose cutoff frequency lies near $M/(4T_S)$. In doing so, a quaternary (M=4) signal is attenuated for symbol values ±3, but not for symbol values ±1, thereby amplitude-modulating the signal. With this modulation, the digital-square-and-filter method (4.12) is also possible. The interpolation module from Sec. 4.3 applies to CPM regardless of timing recovery method.

5.3.2 Linear equalization and block-mode CPM receiver

After the previous stages, the received signal has been separated into two complex baseband waveforms (one per SOP), each with two samples per symbol, in sync with the transmitter. The final task for the demodulator is to compute ML decisions on the data symbols, under observation of N symbols. However, the channel filter distorts the transmitted signal so that it no longer adheres to the phase states and phase transitions that are expected of an unfiltered transmission, even with modest values of h. The receiver therefore requires the use of an equalizer to mitigate ISI and conform to the anticipated behavior.

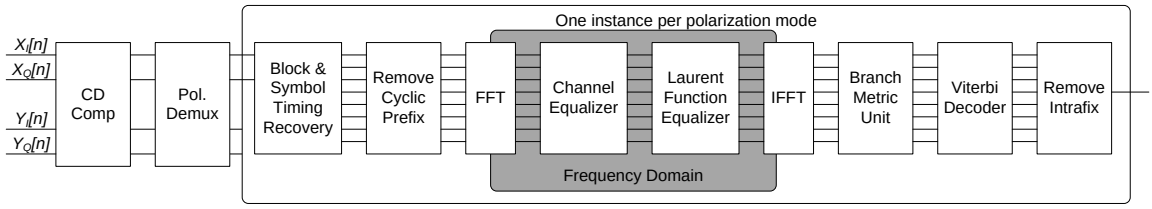


Fig. 5.10. POLMUX block CPM receiver architecture [65]

To accomplish channel equalization, there are two alternative methods. The first, that of linear MMSE equalization, is preferable for signals which pass through the channel filters without significant loss in the sidelobes. The other method is decision feedback equalization, which is much more computationally intense, but enables spectral regrowth for signals that have undergone significantly attenuation in the sidelobes. As a rule of thumb, signals with 98% or greater power-in-band (with respect to the channel filter) should use a linear equalizer, and signals with 95% or less power-in-band will require decision feedback equalization for optimal results. Schemes lying in between can trade off between complexity and performance to determine the appropriate method.

The linear equalization method, enabled by the block-based CPM reception technique [42], makes use of OFDM-style block processing in a polyphase linear algebra space. Prior publication of this technique covered binary CPM techniques, both full response and partial response, but did not apply to *M*-ary signals. In this work, the method has been extended to cover *M*-ary full response signals [65], with full details provided in Appendix B to maintain brevity in this section. Figure 5.10 depicts a diagram of the block-based CPM receiver. In this work, the block-CPM receiver is used for quaternary formats with $h \leq 1/4$, which corresponds to a power-in-band of roughly 98.5%.

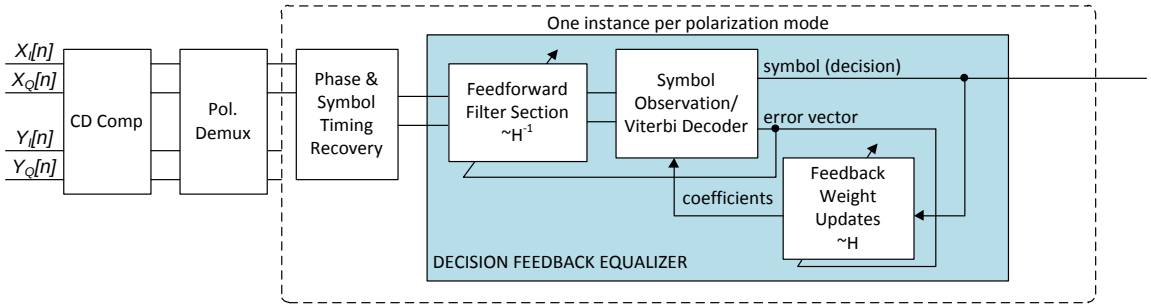


Fig. 5.11. Decision feedback equalizer receiver architecture

5.3.3 Decision Feedback Equalizer

The linear equalizer structure for CPM is inadequate for formats with significant cutoff of high frequency signal content, as mentioned in the previous section. One alternative approach is to use an adaptive DFE, where previous decisions are essentially re-modulated at the receiver through an estimated channel filter to emulate their influence on the current symbol. Future symbols beyond the current symbol (i.e. post-cursor) are also fed through the equalizer to assist with the current decision, though their contribution is limited to that of a linear filter. Final decisions for symbol values are not taken until the N symbol observation window is complete, so therefore there are M^N possibilities over which the equalizer must operate. A block diagram of this architecture is illustrated in Fig. 5.11.

This equalizer structure essentially acts like a maximum likelihood sequence estimator (MLSE) receiver, except that it adaptively estimates the channel as decisions are made. In fact, a full implementation of this receiver type is likely to benefit from invoking a single instance of the DFE receiver type to estimate and track the channel ISI over one block of the received data at a time, while the remainder of the blocks simply uses the channel estimate to fill the MMSE trellis parameters. Since the use of an N-symbol observation window implies the use of Viterbi decoder, regardless of equalizer, a

joint DFE/MMSE receiver is a good choice for quaternary formats with $h > 1/4$ where linear equalization is not beneficial.

CHAPTER 6

SIMULATIONS AND EXPERIMENTS

Previous chapters outlined the fundamentals of fiber-optic systems, CPM and other phase modulated formats, and identified a range of schemes of interest for high-speed transmission. In this chapter, select results from several simulated and experimental studies are reviewed to quantify the behavior of CPM, QPSK, and OQPSK across fiber-optic links. Throughout the discussion the objective is to use QPSK as a baseline format to which the alternative methods are compared, given that QPSK is a standard 100 Gb/s format with more readily available components for implementation.

First covered are the CPM results, which are confined to simulation due to the lack of necessary hardware in the experimental testbed. Secondly, experiments comparing QPSK and OQPSK are covered, and finally simulations of different QPSK schemes in the 1310 nm band are reviewed. The chapter is concluded with a discussion of nonlinear effects in the context of PAPR, giving insight to a general framework by which to anticipate nonlinear behavior.

6.1 CPM Simulations

Since CPM is relatively difficult to implement, the purpose of simulating CPM in a fiber-optic environment is to determine whether any benefit can be gained by selecting it instead of QPSK or another format. The original hypothesis is that the constant intensity feature of CPM is beneficial for avoiding nonlinear effects in fiber. However,

the constant intensity feature does not necessarily stay intact when the transmitted signal is heavily filtered, as described in Sec. 5.1. In fact, those schemes in which it was possible to achieve better minimum distance in a linear channel do so with a significant amount of power in the sidelobes attenuated by the channel filter. Due to the vast parameter space, high cost of implementation, and uncertainty of behavior, simulation was indeed the only practical choice for this study. Given the closeness of matching between simulation and experimental studies achieved by the 100G Consortium [66], there is reasonable confidence that the trends observed in simulation will exist in corresponding experiments.

6.1.1 Simulation environments

Three simulation environments were used for studying CPM formats in fiber. The first is a simple back-to-back setup for isolating linear behavior of the modulation formats (i.e. varying ASE for BER vs. OSNR evaluation). The second environment used a single span of uncompensated fiber and varying launch power, thus isolating XPM and SPM from ASE. The third environment is a multi-span link with EDFAs and dispersion compensation per span, the purpose of which is to quantify performance over realistic deployment scenarios.

The back-to-back environment, depicted in Fig. 6.1, is the most fundamental setup for evaluating modulation formats in a linear noise environment without additional fiber propagation effects. The environment comprises the following components (with corresponding variable parameters):

- Modulation format via custom Matlab module
- Pseudo-random data source (pattern length and generator polynomial)

- Laser source (linewidth, power level)
- Quadrature double-nested MZM modulator (extinction ratio, bandwidth)
- Transmitter channel filter/arrayed waveguide (transfer response)
- ASE noise source (to vary OSNR)
- Receive optical filter (transfer response)
- Coherent receiver (relative intensity noise)
- Balanced photodiodes (bandwidth, responsivity)
- TIAs (gain, gain imbalance)
- Entirely custom Matlab demodulator component

Each component can be independently varied, allowing for a thorough examination of its

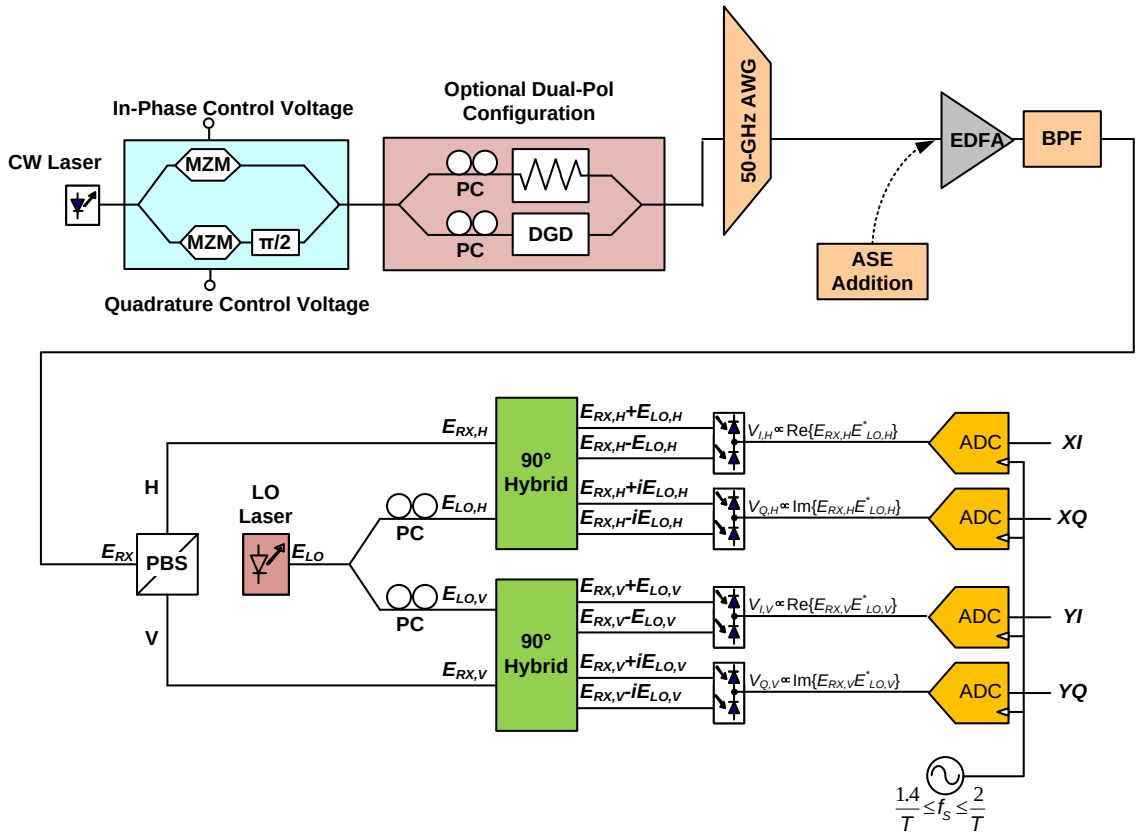


Fig. 6.1. Back-to-back test environment

effect on the modulation format in question. For example, laser linewidth can be set to zero to simplify phase recovery while evaluating other aspects of the demodulator.

The second simulation environment (Fig. 6.2) includes the entire back-to-back setup, adding transmitters as well as a single span of fiber. The corresponding parameters for these new components are:

- Side channel transmitters (format, channel spacing, state of polarization)
- Fiber (length and type)

The link contains no in-line optical dispersion compensation, which must be performed in the demodulator DSP code. This environment is useful for isolating nonlinearities in the span while maintaining high OSNR. Furthermore, since nonlinear coefficients depend upon fiber type, it enables the study of the efficacy of the formats in various fiber types.

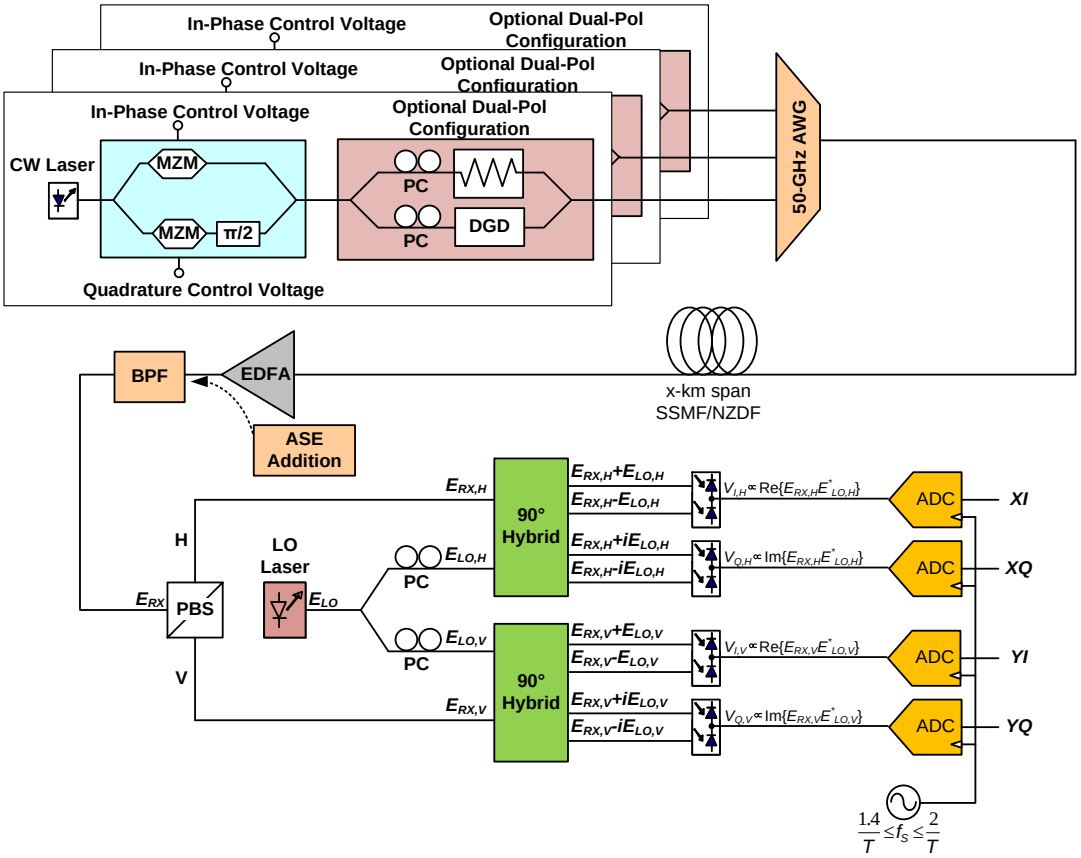


Fig. 6.2. Single-span test environment, featuring ASE noise addition at the receiver

The third simulation setup in use (Fig. 6.3) expands the above to multiple spans of fiber with optional in-line dispersion compensation and EDFA gain stages. These components are parameterized as follows:

- In-line DCF (pre-comp., post-comp., percentage of residual CD comp. per span)
- EDFA (noise figure, gain)

This setup resembles a realistic network deployment, enabling the aggregation of all linear and nonlinear effects incurred along such a link. Optimizing performance in this scenario is the main objective, prior to attempting experimental evaluations of these systems.

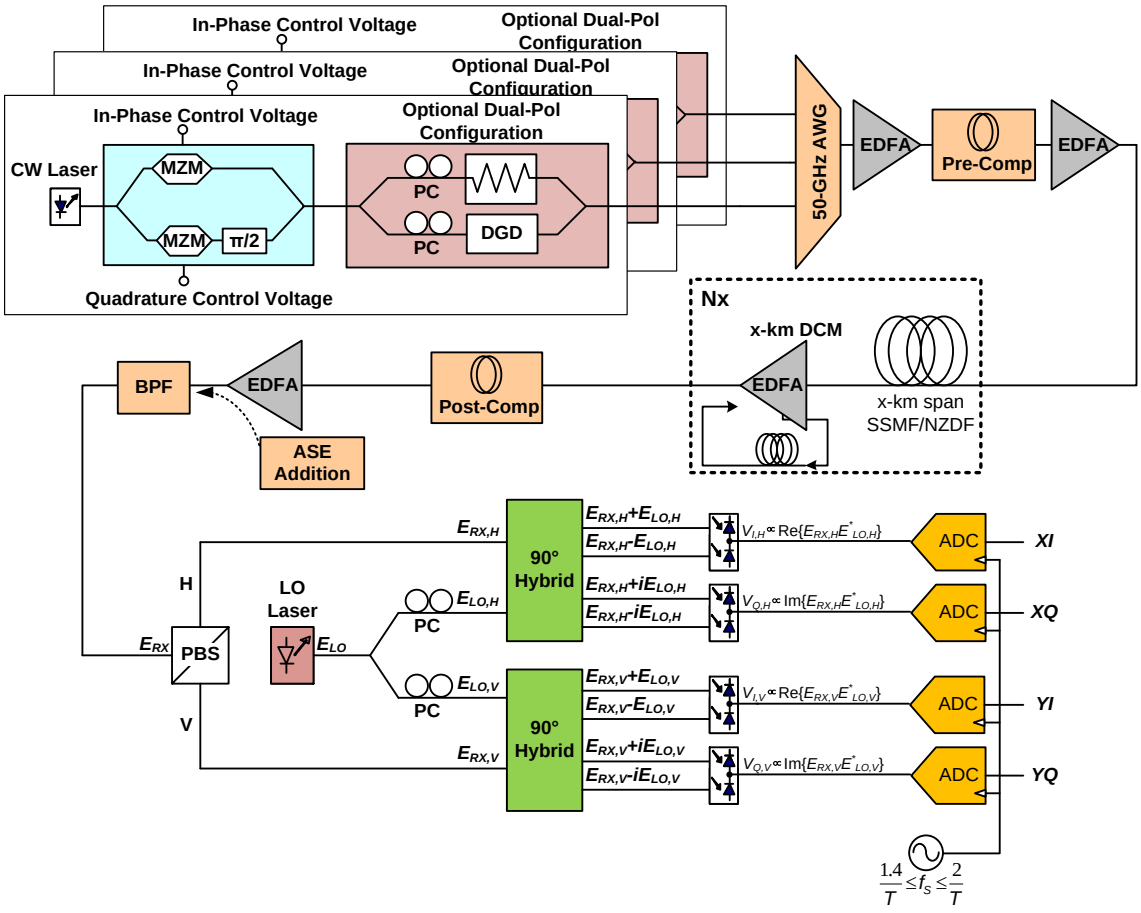


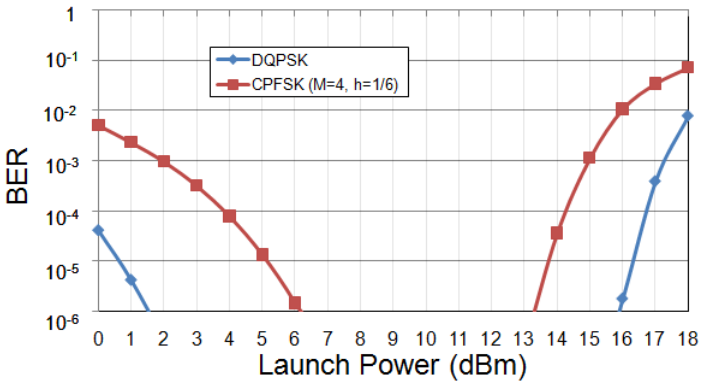
Fig. 6.3. Multi-span test environment, featuring ASE noise addition at the receiver

6.1.2 Single-span tests

As a proof-of-concept, the first set of simulations was conducted with the single-span setup (Fig. 6.2). The purpose was to discern any difference in resilience to fiber nonlinearities between one select CPM scheme and QPSK [67]. In this simulation, a single 90 km span of AllWave ZWP (SSMF class) or TrueWave RS (NZDF class) fiber. The candidate scheme was CPFSK (1REC) with $h=^1/_6$, chosen since its spectrum resides almost entirely within the AWG passband, and therefore remains nearly constant intensity at the fiber entry point. This scheme is compared to a differentially encoded RZ-QPSK transmission, with three identical modulators of either type at 50 GHz spacing launched in the fiber at the same initial state of polarization. In a coherent system, the CPFSK scheme is expected to operate at roughly 5 dB penalty to differentially encoded QPSK (see Fig. 3.4c and [36]). To simplify the evaluation, laser linewidth is set to zero for both formats. Both formats were processed through a coherent receiver, with DDE employed for QPSK and block-mode reception with linear equalizer for the CPFSK signal.

The first trial in this setup was over 90 km of AllWave ZWP (SSMF) in the 1550 nm band. This band is preferred for long haul transmission due to its low loss (0.19 dB/km), the availability of EDFAs, and is characterized by moderate dispersion (17 ps/nm-km) and a relatively low nonlinear coefficient. The resulting BER vs. launch power curves are plotted in Fig. 6.4., indicating that the CPFSK scheme operates at a penalty compared to the QPSK scheme throughout the entire range of launch powers. In the linear region (low launch power), the CPFSK scheme operates at roughly 5 dB penalty, and as nonlinearities take over (high launch power) it operates at roughly 3 dB

(a) 1550 nm



(b) 1310 nm

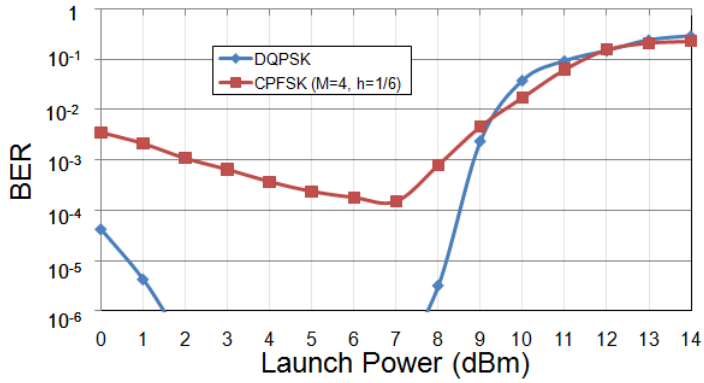


Fig. 6.4. Single-span results for SSMF in the (a) 1550 nm band with 17 ps/nm-km dispersion, and (b) 1310 nm band with 0.5 ps/nm-km dispersion.

penalty. The same setup was used for a study at 1310 nm, where there is slightly higher loss (0.32 dB/km) and negligible dispersion in the fiber (0.5 ps/nm-km). This case has significantly different results, where the CPFSK scheme is incapable of operating at a low enough BER to be feasible, even with enhanced FEC. On the other hand, the QPSK scheme is much more resilient to effects in the fiber.

A further set of simulations were performed over fiber modeled with TrueWave RS (NZDF) parameters, while all other aspects of the previous trial were maintained. NZDF fiber has similar loss (0.20 dB/km) and lower dispersion (~4.5 ps/nm-km) in the 1550 nm band and higher nonlinear coefficient than SSMF. The results (Fig. 6.5) reiterate the 5 dB penalty in the linear regime; however in this case indicate a range in the nonlinear regime where the CPFSK scheme outperforms the QPSK scheme. Though better performance for both formats can be achieved at lower launch power, the

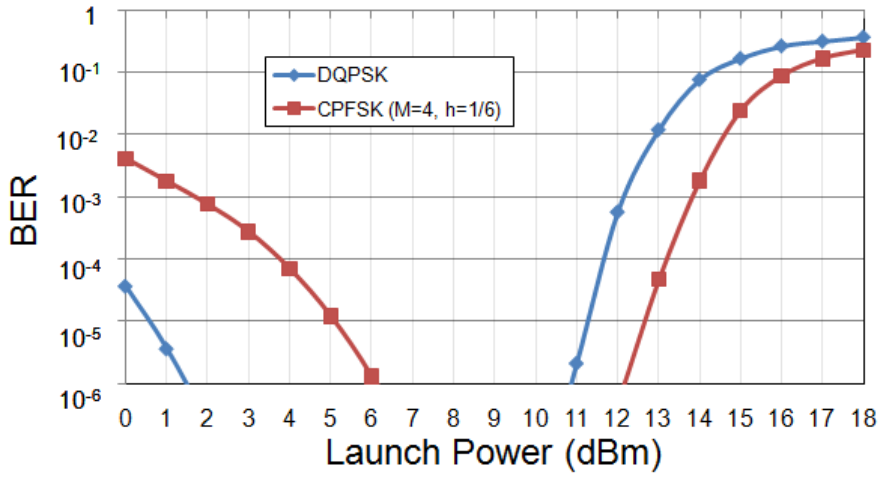


Fig. 6.5. Single-span results for NZDF in the 1550 nm band with 4.5 ps/nm-km dispersion.

interaction between CD, intensity fluctuation, and fiber nonlinearities shows a higher nonlinear threshold for this particular CPFSK format.

An interesting aspect of these trials is the impact of dispersion on the formats as they propagate. In the more dispersive 1550 nm band, the envelope of the CPM signal is significantly distorted after launch. The impact of dispersion on the standard deviation of intensity is plotted in Fig. 6.6 CPM and QPSK, both in single-channel and five channel scenarios. Although the single-channel CPM signal maintains lower fluctuation, its behavior is similar to QPSK's intensity deviation after roughly 15 km. When multiple channels are launched in a single fiber the power is additive even though they are orthogonal in frequency. The resulting intensity fluctuation is rapid and not widely

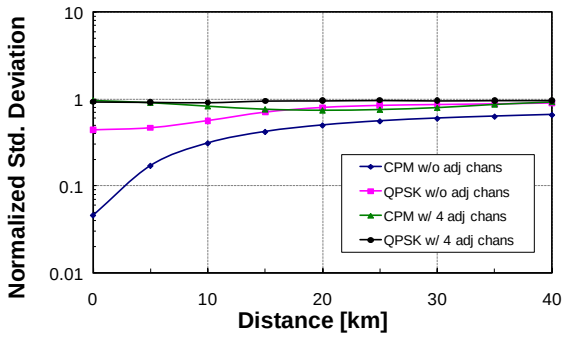


Fig. 6.6. Standard deviation of intensity versus distance in the 1550 nm band in SSMF, normalized for loss along the fiber.

distinguishable between formats. At 1310 nm, the lack of dispersion compounds the impact of nonlinearities since the signals all propagate with zero differential group delay. This partially explains why the three-channel PCM case does not fare well in the 1310 nm SSMF application, even though the individual carriers are constant intensity.

6.1.3 Back-to-back tests

The single span tests provided several insights. Firstly, robustness to nonlinearities depends on modulation format and fiber, namely the interaction between CD, loss, and nonlinear coefficient. Secondly, narrowband CPM schemes operate at a significant penalty to QPSK, except in the nonlinear regime in certain cases. Though not specifically plotted in Figs. 6.4 or 6.5, the minimum BER is achieved with QPSK at a lower launch power than CPFSK at its optimal launch power. Finally, a system with multiple channels suffers from intensity variation as the superposition of intensity as photons of different wavelengths interfere with each other. Thus, the constant intensity of each individual subcarrier is insufficient for eliminating XPM.

Although the above analysis is contrary to the original hypothesis that CPM formats could offer significant robustness to nonlinearities versus other formats, there are at least three potential areas of interest remaining. First is to explore CPM schemes that offer higher minimum distance than QPSK, potentially providing better performance. Second is that the constant intensity feature results in the lowest PAPR possible per carrier. In other words, a CPM scheme will exhibit lower peak power than any other format at the same launch power. Since the peak intensity of an N subcarrier system is N^2 times the peak power of each subcarrier [25], the probability that a WDM link of several CPM carriers will experience an error-inducing XPM phase shift is lower than other

formats. Finally, for single-carrier links the choice of CPM still makes sense for totally avoiding nonlinearities, although such an application is not the focus of the 100G Consortium's research effort.

In Sec. 5.1, the selection of CPM parameters was studied in a DWDM environment. It was concluded that CPFSK schemes offer optimal performance under the SE and ROADM bandwidth objectives of the 100 Gb/s application, with specific interest in schemes with h on the range of roughly $^1/_4$ to $^2/_5$. It was further discussed in Sec. 5.2 that the use of a QM-based transmitter with two samples per symbol in the bandlimited spectrum would minimize implementation complexity, effectively converting the system into a correlative PSK system. The simulation effort therefore turns to evaluating performance of such systems in fiber to discern whether any of the potential performance gain can be achieved in a fiber based system.

The first evaluation is in a back-to-back environment, with a QM-based transmitter and the DFE receiver from Sec. 5.3.3. Since the receiver type is significantly more complex than the block-based receiver, the QPSK format in comparison is processed by a DFE for fairness. A single laser with zero linewidth is used in a single-pol. mode, centered at 1550 nm. The signal is launched through a 50 GHz AWG (the filter response of which was experimentally gathered in the experimental testbed), and passes through a single 40 GHz super-Gaussian filter (order 3.5) prior to reception. No side channels are present. BER vs. OSNR results for the back-to-back setup are depicted in Fig. 6.7. All modulation indices between $^2/_7$ (0.2857) and $^3/_8$ (0.375) are competitive with QPSK in the back-to-back setup, where additive noise is the only impairment. The best format appears to be $h=^5/_{16}$, which has a steep downward slope due to excessive

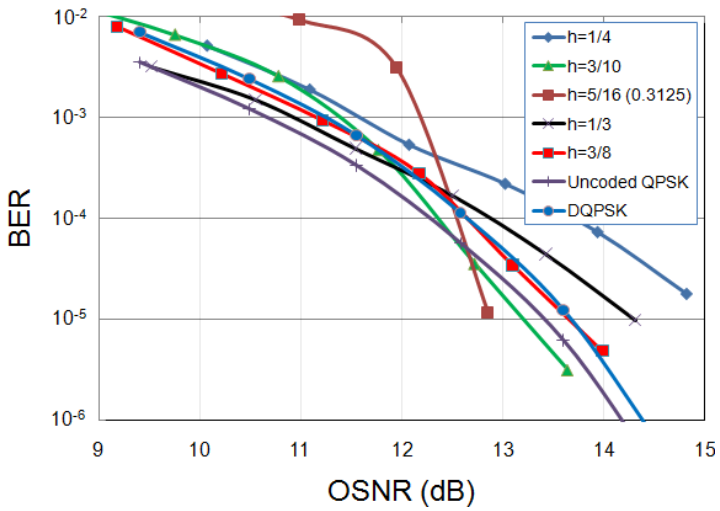


Fig. 6.7. BER vs. OSNR in back-to-back simulations for QPSK and CPFSK signals over a variety of modulation indexes.

errors at the ~12 dB OSNR point. These errors occur when the Viterbi decoder falls off track and has a lower probability of regaining the correct phase state. As a counter-example, the $h=1/3$ system is not particularly prone to these errors since there are only three possible phase states per symbol period, though this also explains why there is lower minimum distance for that scheme (refer to Figs. 3.4c and 5.4a).

6.1.4 Multiple span tests

The back-to-back configuration highlighted a few schemes of interest for further evaluation in a DWDM simulation through spans. In the following simulation, four 90 km spans of TrueWave RS (NZDF) fiber is used, with five co-polarized single-pol channels all operating with the same modulation format. The focus was placed on the $h=1/3$ format due to its relative ease of implementation, again compared to QPSK. All other parameters of the back-to-back test were maintained, though launch power was varied (+1 dBm, +3 dBm, and +5 dBm per carrier). BER vs. OSNR in this study is graphed in Fig. 6.8, and shows that the performance of the $h=1/3$ format is roughly equivalent to QPSK at the lowest launch power, then gains a performance advantage over

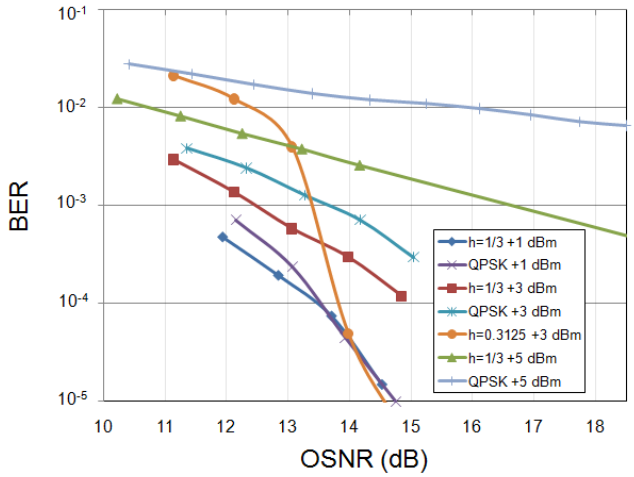


Fig. 6.8. BER vs. OSNR across 4 spans NZDF for QPSK and CPFSK signals over a variety of modulation indexes.

QPSK at the higher lauch powers. In fact, at +5 dBm, the QPSK system was unable to achieve BER as low as the presumed 10^{-3} threshold for enhanced FEC with the ASE source turned off at the receiver, whereas the CPFSK surpassed the target.

In conclusion, there are many manifestations of CPM formats that cover a broad range of SE, minimum distance, and implementation complexity. In general, the usefulness of CPM schemes to compete with QPSK in the 100 Gb/s application over 50 GHz spaced channels is reserved for a few cases where local dispersion is low throughout the link and nonlinearities dominate at the required launch power. The implementation complexity alone mitigates the potential impact of CPM in high speed fiber-optic links, but this assumption may change as future technological developments enable faster processing and higher resolution data conversion. Additionally, future deployment scenarios may reveal that DWDM links in which all carriers are phase modulated are subjected to very limiting total power. In such an event, higher grid spacing among the allowed carriers would enable the passage of higher modulation indexes (e.g. h=0.4) at a more significant gain over QPSK than is available with the narrower channels.

6.2 QPSK and OQPSK Simulations and Experiments

QPSK is the standard format for initial 100 Gb/s deployment [8], and is therefore at the forefront of the 100G Consortium's research. Design and implementation of the 100G Consortium's QPSK demodulation DSP code, as detailed in Ch. 4 and Appendix A, is a major contribution of this dissertation work. The custom Matlab code for receiver algorithms is portable for use between the RSoft environment as well as in the context of offline processing of samples taken from the experimental testbed by one or more real-time oscilloscopes. The coherent receiver and DSP algorithms play an essential role in the PM-QPSK experimental setup, enabling a wide variety of studies for current and future networking technologies.

In addition to the code presented here, many experimental QPSK results have been collected with code from Optametra, the supplier of the consortium's first coherent optical receiver. The Optametra code uses different algorithms for estimating SOP and phase, including estimation methods requiring *a priori* knowledge of the transmitted data pattern. These methods provide excellent performance, and are especially useful when characterizing optimal performance of a link under certain parameters, but are not necessarily practical for deployed systems due to pattern dependence. All results from the Optametra receiver are enhanced by inserting the code developed here to perform CD compensation prior to timing recovery and DDE prior to pattern detection.

Offset QPSK is interesting to this study due to its similarity to CPM and its compatibility with QPSK. An OQPSK signal can be generated using a QPSK modulator by simply delaying either I or Q component by half a symbol period. However, RZ pulse carving is not possible without specifically integrating the pulse carver in each of the I

and Q waveguide paths. And although the demodulation processes differ (see Ch. 4), OQPSK generally follows the same signal flow and requires similar complexity as QPSK.

6.2.1 QPSK experimental results

A variety of experiments have been conducted in the experimental testbed with QPSK modulation of the channel of interest. The two most significant experiments to date are the studies of the crosstalk penalty [33, 34] and the NLT [32]. The former investigates the in-band penalty as channels are added and dropped at ROADM sites in mid-span in DWDM systems. The latter introduces a new metric, the NLT, as a rule-of-thumb for scaling 100 Gb/s PM-QPSK systems.

The experimental setup for the inband crosstalk penalty is the multi-span environment (see Fig. 6.3), where only a single channel was launched, then split and shaped with a WSS to adhere to one of five shaped spectrum profiles. Each crosstalk profile represents a unique channel arrangement in the context of a ROADM, where a channel is dropped and another independent channel is inserted in its place (see Fig. 6.9). The experiment was conducted over AllWave ZWP and TrueWave RS with and without in-line optical dispersion compensation. Experimental results reveal that although the contribution of each crosstalk profile differs (Fig. 6.10a-c), they differ as expected in the sense that the crosstalk power focused at the center of the channel has a greater influence

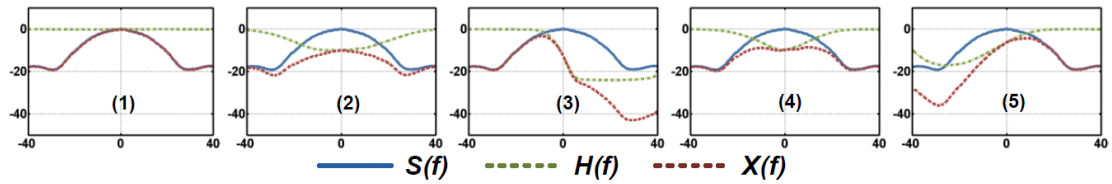


Fig. 6.9. Crosstalk profiles under analysis, each modeled by a WSS to cover different add/drop scenarios. $S(f)$ is the spectrum for the interfering signal, $H(f)$ is the WSS filter shape, and $X(f)$ is the shaped crosstalk $X(f)=H(f)S(f)$. [34]

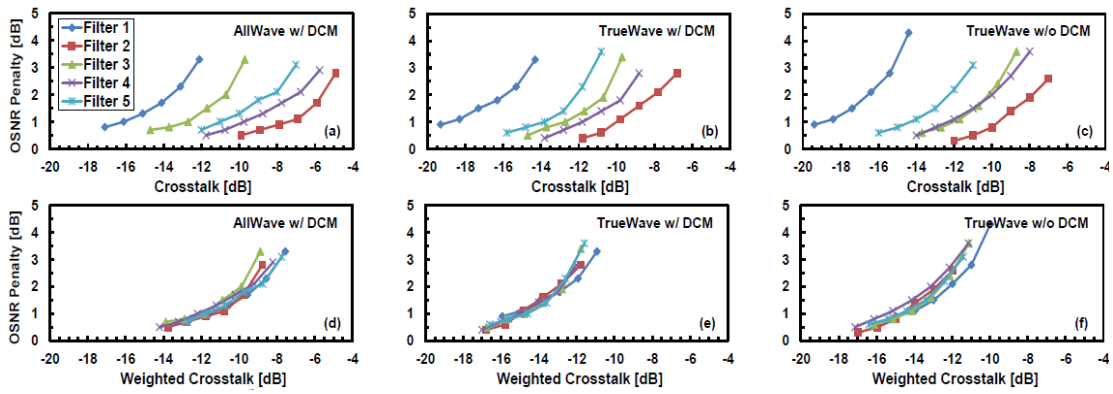


Fig. 6.10. OSNR penalty vs. crosstalk power for five crosstalk scenarios and three link types, showing uweighted (a-c) and weighted (d-f) results [34]

than the same amount of power away from the center. To accommodate for this effect, the notion of a weighting function was developed, where the interference spectrum is first filtered by the spectrum of the signal of interest, then integrated. After the weights are accounted for, the results are virtually independent (<1 dB) of the interference profile (see Fig. 6.10)

In addition to the crosstalk experiments, a study was conducted to determine a scaling metric for performance prediction of a mixed 10 Gb/s OOK and 100 Gb/s QPSK deployment. Once again, the test bed was set up in a multi-span arrangement (Fig. 6.3) with a single QPSK channel with four OOK neighbors in adjacent channels (two on each side). To illustrate this study, consider Fig. 11 which depicts the required OSNR to achieve a BER of 10^{-3} in TrueWave RS fiber with no in-line dispersion compensation. The P_{NLT}, defined as the launch power at which the signal suffers a 1.5 dB OSNR nonlinearity penalty is identified for the case in which the OOK side channels are launched at -4 dBm. Additionally shown is the *XPM offset*, which measures the performance difference in the linear regime of operation between having no sidechannels and having four sidechannels at -4 dBm each.

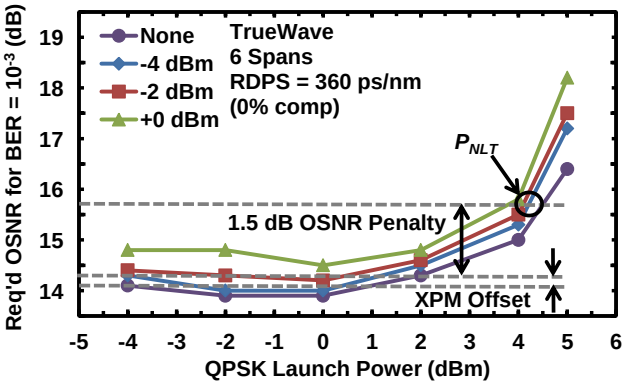


Fig. 6.11. OSNR requirement vs. launch power for QPSK in TrueWave RS without in-line optical compensation, with varying power on the 10 Gb/s OOK side channels (none, -4 dBm, -2 dBm, 0 dBm launch power) [32]. The graph identifies P_{NLT} as well as XPM offset for the -4 dBm sidechannels case.

The results of the study indicate that a robust NLT can be projected for links over which there is no in-line optical dispersion compensation. The NLT is specific to the type of fiber and generally increases with number of spans on a logarithmic basis. The experimental and simulated results for AllWave ZWP and TrueWave RS are shown in Fig. 6.12 for both NLT and XPM offset in each case.

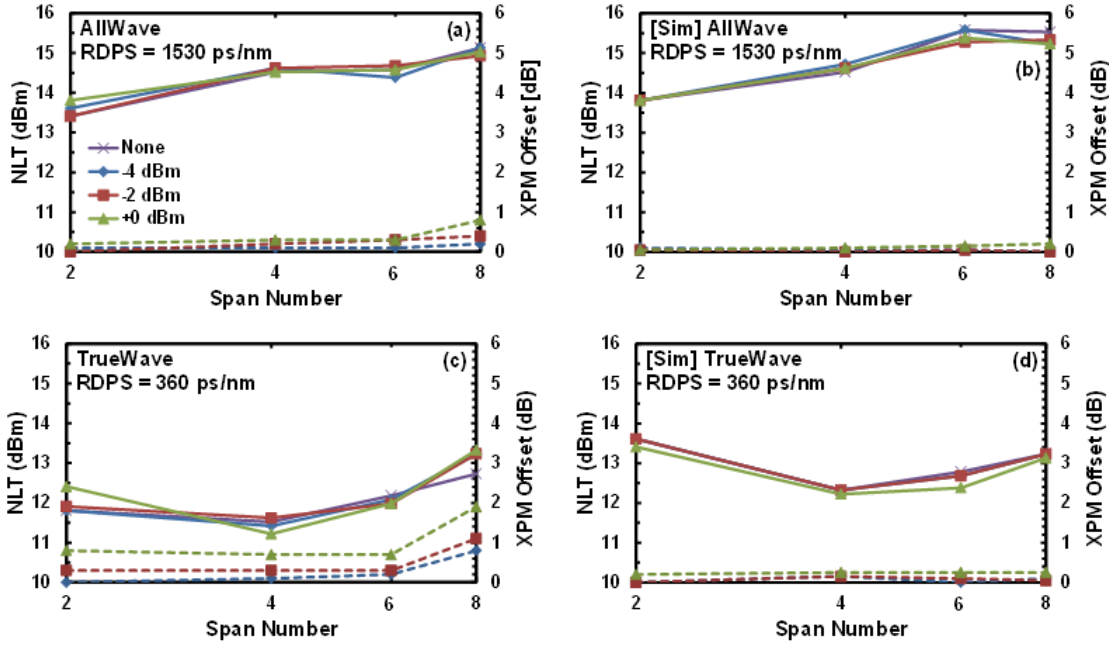


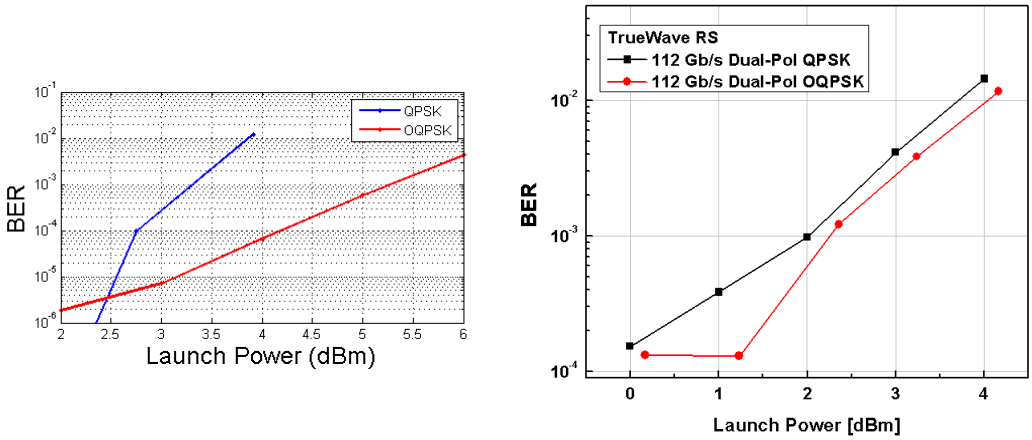
Fig. 6.12. NLT and XPM offset versus number of spans for AllWave ZWP (a-b), and TrueWave RS (c-d) fiber types. Experimental results (a,c) show agreement with simulations (b,d) within 1 dB. [32]

Interestingly, the results for spans with 100% in-line dispersion compensation did not follow the same predictable patterns for NLT or for XPM offset. This was in part due to the accumulative effect of nonlinear phase modulation as the signal propagates. Since there was zero residual dispersion, all pulses were realigned at the launch point of each link. Such a dispersion compensation strategy may be beneficial for OOK networks with a high number of add/drop nodes in which zero local dispersion reduces the overall component count, but is a poor choice for phase modulated systems.

6.2.2 OQPSK experimental results

Using a four-span TrueWave RS and six-span AllWave ZWP testbed (Fig. 6.3), OQPSK was compared to QPSK to determine whether any performance advantages were possible. The setup included 100 kHz linewidth lasers for transmit and receive local oscillators. Preliminary results indicated that OQPSK outperforms QPSK as launch power increases [68, 69], similar to the CPFSK simulations. However, dual-pol results were just slightly lower performance than QPSK (see Fig. 6.13). Overall performance for OQPSK was limited by the NDA phase recovery method used, which achieved optimal performance with an averaging filter spanning roughly 80 symbols. By comparison, NDA phase recovery in QPSK achieves optimal performance under the same link conditions by averaging over 15 symbols or so.

The disparity between the two filter lengths is due to the fact that the OQPSK NDA recovery process is borrowed from the minimum shift keying (MSK) design (Sec. 5.3.1), where the phase states alternate between being located on the I axis and the Q axis. In this experiment only NRZ-OQPSK was possible, which features off-axis locations for symbols whenever the opposite component (in-phase or quadrature) was



(a) (b)

Fig. 6.13. Experimental results for OQPSK vs. QPSK across three adjacent 50 GHz channels across four spans of TrueWave RS fiber in (a) 56 Gb/s single polarization and (b) 112 Gb/s polarization multiplexed configurations.

consistent across adjacent symbols. Better performance may therefore be possible with decision-directed phase recovery.

6.3 QPSK Format Comparison in the 1310 nm Band

The 1310 nm band is characterized by zero dispersion in SSMF, and is specified for use in several IEEE 802.3 Ethernet standards [70], including 10GBASE-LR, 40GBASE-LR, and 100GBASE-LR. The 1310 nm band operates at higher loss (0.32 dB/km in AllWave ZWP) and unlike EDFAs for 1550 nm band, there is no commercially available fiber amplifier technology. Due to these attributes, the 1310 nm band is typically used for SMF applications requiring moderate reach (<40 km), and relatively low cost. Note that transceivers in the 850 nm band over multi-mode fiber (MMF) are used exclusively for the shortest links and lowest cost.

All optical Ethernet formats are currently defined for IMDD systems up through and including 100GBASE-LR4 and 100GBASE-ER4. Scaling the standards to 40 Gb/s was achieved across four bonded carriers, each at 10 Gb/s payload rate. 100 Gb/s is

achieved by scaling each of the four carriers to 25 Gb/s payload rate. Scaling to 400 Gb/s and 1 Tb/s are likely to require change of modulation format to QPSK to increase SE and limit the number of independent carriers for the transceiver to process. The move to QPSK must be cost effective and power efficient, since operators could otherwise choose to use more fiber and transceivers of lower-rate interfaces. The concept of "cost per bit" captures this requirement, balancing the overall bit rate with the cost of the transceiver, fiber, and power.

Though QPSK is the next logical progression for optical Ethernet transmission, three fundamental questions remain specific to the implementation. First is whether a coherent receiver is a good choice in the 1310 nm band, and second is the transmitter architecture that best suits the format, and third is the number and arrangement of carriers. The use of coherent receiver in this case does not seem to be beneficial due to the existence of an adequate differential receiver (see Sec. 2.3), which can achieve the required performance without incurring the additional hardware and power to fully digitize and process the received signal. Recall that in the 1310 nm band, there is a general lack of optical amplification, which limits additive noise besides limiting reach.

To investigate the remaining questions, a single span simulation was performed (Fig. 6.2, without EDFA and ASE noise source) [71]. In this configuration, there were ten individually modulated QPSK carriers, each modulated in single-pol mode at 56 Gbaud. The subcarriers were launched either all co-polarized, or else interleaved between two orthogonal linear states of polarization. Polarization multiplexing was not considered since the WDM configuration is not dense, and it would unnecessarily add complexity to the receiver. Four different modulation formats were chosen for this

examination, NRZ-QPSK, RZ-QPSK, phase modulated QPSK (ϕM-QPSK), and π/4-shifted QPSK. Figure 6.11 shows transmitter block diagrams for each format. Note that OQPSK was not examined since an appropriate differential receiver does not exist.

The ϕM-QPSK format is generated using only two optical PMs, one biased to achieve a π phase shift, and the other for an independent π/2 phase shift, the output of which is constant intensity. This format represents a transition-restricted 7-ary CPFSK format, $h=\pi/2$, as a memoryless variation of CPM. The ϕM-QPSK transmitter is the least complex since it requires only two PMs (recall each MZM is a pair of PMs in parallel), no PC, and no 90° phase delay (and accompanying bias controller). The modulator structure is depicted in Fig. 6.14 along with the conventional structure for comparison. For π/4-shifted QPSK, the pulse carver typical of RZ-QPSK was replaced by a phase modulator to minimize implementation complexity. It is noted however that this strategy for π/4-shifted QPSK requires a narrow channel filter to achieve the desired signal transitions. Reception of π/4-shifted QPSK differs from the other three in that the branch delays (Fig. 2.X) are 0 and π/2 instead of ±π/4.

The goal for the simulations is to achieve the required 10^{-12} BER with the longest possible reach and minimal system complexity. The use of some moderate FEC (e.g.

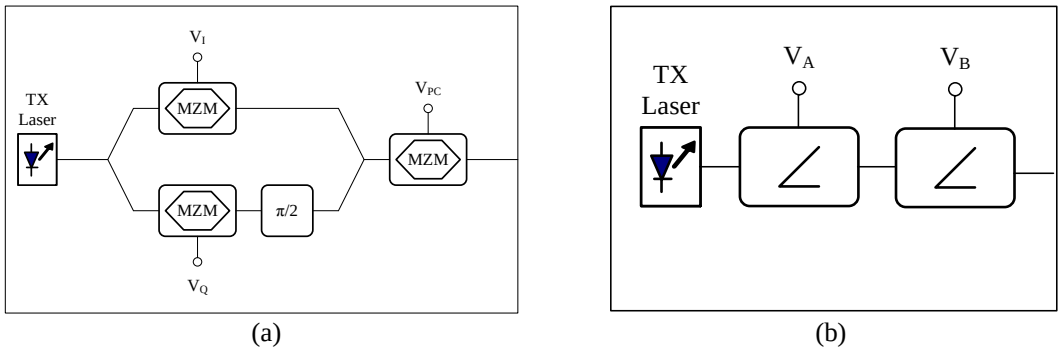


Fig. 6.14. Modulator structures for QPSK transmission for (a) RZ and NRZ formats and (b) for ϕM-QPSK format.

Reed-Solomon coding) may be allowable at farther reaches, although it is noted that FEC exists in telecom OTN standards, but not currently for Ethernet PCS/PMA. The 1310 nm xBASE-LR formats are all specified for 40 km reach; results for 60 km and 80 km spans are shown in Fig. 6.11 and 6.12, respectively. Firstly, the $\pi/4$-shifted QPSK format achieves the worst performance of the schemes through either length of fiber, and can be eliminated due to its excess complexity beyond NRZ. Secondly, there is a huge advantage in launching the carriers with alternating SOPs, which has the effect of minimizing XPM effects (recall that XPM is 50% stronger for co-polarized signals than orthogonally polarized). Therefore, careful consideration to transmitter waveguide and laser alignment is definitely indicated.

Among the other three formats, all are capable of meeting the performance goal at 60 km. NRZ achieves the best performance, though very similar to RZ, and ϕM-QPSK is the worst. Going from the 60 km span to the 80 km span, the optimal launch power for each format increases into the more nonlinear regime to counteract loss. More interestingly however, is that the optimal launch power for RZ is lower than that of NRZ, which in turn is lower than that of ϕM-QPSK. Based on this study, the recommendation would be to use NRZ-QPSK with the minimum number of channels required to reliably achieve 400 Gb/s or 1 Tb/s, and with each alternating channel launched with orthogonal SOP to its adjacent channels. The phase-modulator strategy will likely suffice for 40 km reach at a lower cost, though its performance is more dependent on component and filter bandwidth than NRZ.

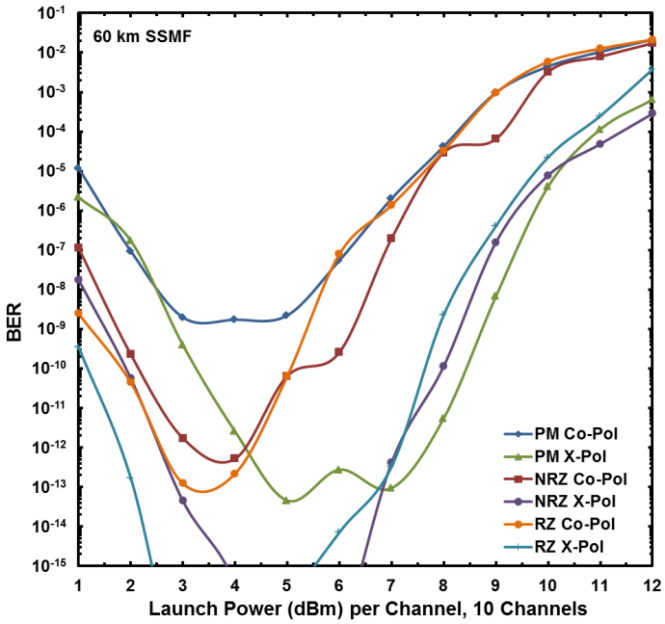


Fig. 6.15. BER vs. launch power for QPSK over 60 km SSMF in the 1310 nm band, covering RZ-, NRZ-, and ϕM-QPSK formats, launched either fully aligned in polarization or interleaved. [71]

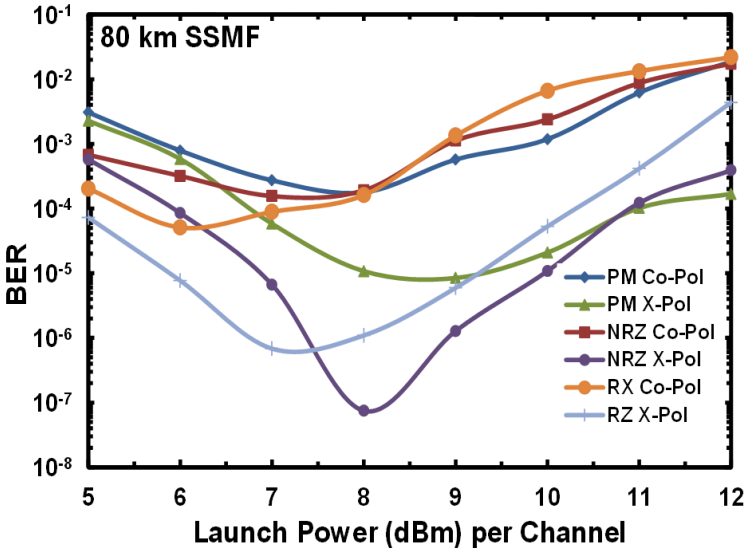


Fig. 6.16. BER vs. launch power for QPSK over 80 km SSMF in the 1310 nm band, covering RZ-, NRZ-, and ϕM-QPSK formats, launched either fully aligned in polarization or interleaved. [71]

6.4 Fiber Nonlinearities and Peak Power

Fiber nonlinearities play one of the most significant roles in limiting channel capacity, as described in Ch. 2. Though SPM and XPM describe intra-channel and inter-channel effects, it is the superposition of all WDM channels within a fiber that give rise to the highest intensity peaks. Intensity peaks form and disperse depending on the CD of the fiber and signal states (amplitude and phase) of the underlying carriers as the signals propagate. Importantly, the influence of these peaks is significant only along the first 10-20 km of fiber per span (the loss-dependent effective length of fiber). If possible, minimizing nonlinear effects in fiber is a matter of reducing the intensity of the peaks formed by the superposition of all channels together. In a wireless OFDM application, this particular problem is referred to as the PAPR reduction problem [25].

PAPR in OFDM results when a pattern forms in the Fourier space of the transmitted subcarriers. One such pattern could be when all N subcarriers are transmitting the same complex-valued symbol. In OFDM, the transmit power (here forms the "average power" of the ratio) is fixed, and peaks that result at the output of the transmitter's IFFT may be higher than the range of the output DACs or outside the linear range of the subsequent amplifier. Any clipping or distortion of the signal may result in errors at the receiver, and therefore several methods have been developed to reduce PAPR in OFDM systems.

For any single carrier system with transmitted signal $s(t)$, let the peak signal amplitude be denoted as

$$A_{\max} = \arg \max_{t} \left(s(t) \right) . \tag{6.1}$$

The maximum power is therefore $P_{max}=A_{max}^2/t$. The maximum possible amplitude from the superposition of N identical carriers is $N \cdot A_{max}$, with a corresponding power $N^2 \cdot P_{max}$. In a single-mode fiber optic system, the two linear polarization modes add independently from each other. Hence, in an N-carrier single-pol system the worst case peak along each SOP is $N^2 \cdot P_{max}$ if all carriers are co-polarized, and $(N/2)^2 \cdot P_{max}$ if the carriers are evenly divided over two orthogonal SOPs. The nonlinear phase shift is proportional to intensity (2.15), so the role that constant intensity plays in nonlinear penalty reduction in WDM systems is not to eliminate XPM. Rather, the role is to achieve the lowest probability that the aggregate signal intensity surpasses some level that carries the phase modulated signals away from the decision region.

To illustrate the influence of PAPR in WDM fiber systems, consider Fig. 6.17 which shows a simulated single-carrier signal for the three QPSK variants in the 1310 nm QPSK study. Note that the average power of each plotted signal is identical, but the peak amplitude for RZ is greatest, NRZ is lower, and ϕM is minimal. (Although the NRZ signal appears to have similar peak power to ϕM at this launch power, some of its power is wasted in bowing 90° and 180° transitions due to the non-ideal extinction ratio of the

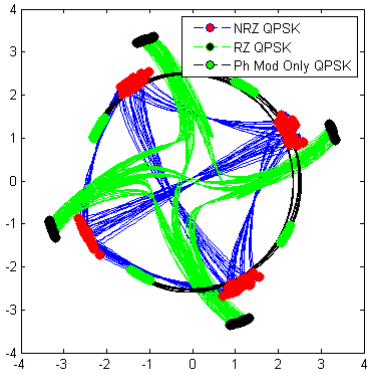


Fig. 6.17. Signal constellation and transitions for the optically generated QPSK formats RZ-, NRZ-, and ϕM-QPSK, all at +8 dBm launch power.

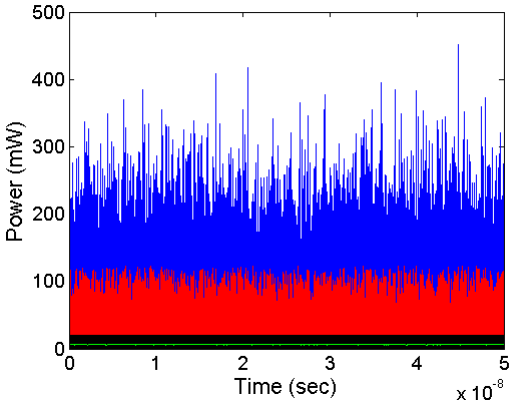


Fig. 6.18. Instantaneous optical power for co-polarized signals, all modulated by the ϕM-QPSK transmitter and filtered (400 GHz) prior to the combiner: green – 1 channel, black – 2 channels, red – 5 channels, blue – 10 channels.

MZMs.) At fixed launch power, it therefore appears that a differential receiver (inherently incapable of tracking symbol transition behavior) would benefit from the use of RZ pulse carving to maximize the available minimum distance.

Moving from a single carrier to multiple carriers, the time-domain progression of the composite multi-carrier signal is much more rapid than that of each individual carrier. The composite signal may, in fact, be viewed as a single carrier encompassing all the independently modulated carriers, albeit with a unique modulation format. Such a system module does not require the carriers to be modulated with the same format, or even at the same symbol rate. Importantly, however, is that the amplitude and duration of power peaks of such a system depend on the modulation format and rate of the carriers, the number of carriers, and the power level of each carrier. To illustrate this point, ϕM-QPSK data from the 1310 nm simulation is shown in Fig. 6.18 for one, two, five, and ten carriers, each launched at +8 dBm. The single carrier has virtually no fluctuation in its intensity, since the phase modulator is incapable of attenuating optical power and the transmit filter was chosen to be wide enough (400 GHz) to attenuate very little of the modulated signal spectrum. However, a plurality of carriers results in instantaneous

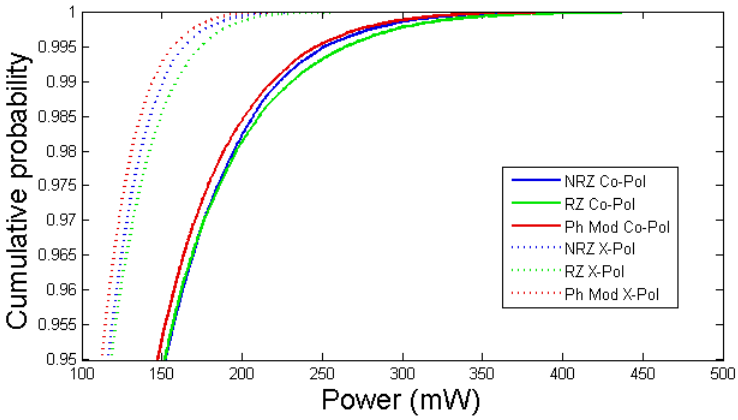


Fig. 6.19. Cumulative probability density function of sampled instantaneous power in the ten-channel 1310 nm single-link transmission for RZ-, NRZ-, and ϕM-QPSK, launched with full alignment in polarization modes (Co-Pol) or interleaved (X-Pol). [71]

intensity peaks, even with only two carriers. Figure 6.18 indicates that the peak power level grows as the number of channels increase while the probability of reaching the maximum peak decreases.

Nonlinear phase modulation is proportional to instantaneous intensity. For PSK, CPM, QAM, and OFDM formats, errors will occur if the nonlinear phase modulation term exceeds the angular offset separating the decision regions (or less, in the presence of additive noise). Across different formats with the same minimum distance (e.g. RZ, NRZ, and ϕM-QPSK), the probability with which the intensity is high enough to produce an error-inducing phase shift determines the BER. The cumulative distribution function (CDF) of instantaneous power is shown in Fig. 6.19 for these three modulation formats in the 1310 nm study of Sec. 6.3. All are launched at +8 dBm, and both polarization scenarios are plotted.

The most noticeable attribute from Fig. 6.19 is that co-polarized signals (including dual-pol operation) suffer significantly higher peaks, again because photonic interference is dependent upon SOP. In other words, individual photons are polarized and will interfere only with photons which share (partially or fully) the same SOP.

Among each SOP configuration, there are differences between the three modulation formats. The results here are consistent with the singl-channel constellation in Fig. 6.17, where for almost every instantaneous power level, ϕM-QPSK is least likely to reach the level, NRZ is next, and RZ is most likely.

The body of simulations and experiments has shown that the severity of nonlinear impairments depend upon the interaction of modulation format, SOP, number of subcarriers, loss, CD, and of course on the nonlinear coefficient of the underlying fiber. The impact of these parameters is as follows:

Modulation format, SOP, and number of carriers: Peak power at the fiber launch along each SOP results from the superposition of all carriers with photons aligned to the SOP. The number of independent carriers and the peak power of each determines the peak power, and therefore the severity of nonlinear phase shift.

Loss: The fiber effective length is the distance over which the signal power is high enough to have a noticeable impact on nonlinearities. Higher loss inhibits nonlinear interactions, low loss extends them.

Chromatic dispersion: Peak pulses will form and disperse throughout propagation in the presence of non-zero CD. In the 1310 nm SSMF application or ~100% in-line compensation (i.e. 0 RDPS), the initially launched signal with its peaks will dominate nonlinear phase modulation. For uncompensated links, or those with non-zero RDPS, pulse formation and dispersion becomes independent of modulation format throughout propagation, causing intensity to resemble a Rayleigh distributed random process.

Nonlinear coefficient: The nonlinear coefficient determines the relative severity of nonlinearities, and therefore limits the available launch power amongst all channels in a fiber.

In light of these results, the following conclusions can be made regarding the use of CPM in fiber-optic links:

1. Constant intensity of a CPM carrier is impacted by the optical channel filter, and a tradeoff exists between maintaining constant intensity and maximizing minimum distance in a bandlimited channel.
2. Constant intensity is a significant advantage for mitigating SPM, making it a compelling format for single-carrier communication (i.e. not WDM) in a zero-dispersion propagation mode, and somewhat beneficial for a ~100% (~0 RDPS) in-line compensated link.
3. In a multi-carrier environment (i.e. WDM), the resulting peak intensity is minimal for CPM compared to other formats, but does not provide the same impact as in a single-carrier environment. Like the single-carrier case, the dispersion map of the link will determine the extent of benefit this will provide to avoiding nonlinear phase modulation.
4. Conversely, in a highly dispersive environment, especially one with multiple amplified spans, the constant intensity of a single CPM carrier will be prone to a similar amount of nonlinear impairments as any other modulation format launched at the same power. Thus, performance will depend entirely upon minimum distance and implementation loss in the system.

5. If an application requires SE of ~1.4 b/s/Hz or greater, the available minimum distance gain over QPSK does not justify the additional complexity of implementation. On the other hand, certain fiber types with low dispersion and high nonlinearity will accommodate fewer phase-modulated carriers at a given power level than highly dispersive links. Such an application may permit the use of higher channel spacing and bandwidth, and hence lower SE.

Therefore, the use of CPM in high speed fiber-optic links will likely be limited to highly nonlinear, low dispersion fiber that would otherwise be incapable of accommodating a large number of phase modulated carriers.

CHAPTER 7

CONCLUSION

7.1 Summary

State of the art fiber-optic systems are transitioning towards phase modulated formats to enable higher spectral efficiency and scale to faster data rates. Polarization-diverse coherent receivers further increase the spectral efficiency by enabling polarization multiplexing and allowing arbitrary choice of modulation format. Additionally, the use of enhanced FEC greatly increases the allowable raw BER. Altogether, these advances represent a revolutionary change in the technology, significantly enhancing reach and data rate.

The use of phase modulated coherent optical carriers alters the focus of mitigating fiber impairments. Whereas dispersion (CD and PMD) played a major role in IMDD systems, they can be fully compensated by DSP within a coherent receiver. On the other hand, nonlinear phase and polarization modulation play a critical role in determining available launch power, number of allowed WDM carriers, reach, OSNR, and ultimately BER. These nonlinear interactions arise from fluctuations in intensity of the underlying optical signal. It is therefore necessary to understand these limitations and identify means for mitigating them.

CPM is a class of phase modulation formats that differ from other schemes in that the signal amplitude is constant throughout transmission, even during symbol transitions.

A CPM format is characterized by several different parameters, which together determine the SE, minimum distance, and implementation complexity of the system. Generation of an optical CPM carrier can be accomplished through a novel QM/PM structure, or alternatively by a DAC/QM with oversampling of the desired signal. Reception is compatible through a polarization-diverse coherent optical receiver, with DSP for performing demodulation.

Coherent optical reception, as a building block for the next generation of fiber-optic links, encompasses a variety of demodulation functions required to recover the transmitted data. These functions include chromatic dispersion compensation, polarization demultiplexing, timing and carrier recovery, channel equalization, and symbol decoding. The choice of implementation of each unit is governed in part by the applicability to real-time processing at or above 28 Gbaud. Therefore feedforward implementations that readily parallelize for block processing are generally favorable. Feedback functions, such as channel equalization, are permissible if they are capable of burst-mode operation, where states at the end of a block will be appropriate for use at the outset of a subsequent, but not contiguous, block of samples (or symbols). Based on this methodology and set of assumptions, a set of QPSK demodulation code was developed as part of this effort for use in the 100G Consortium's experimental and simulation efforts (presented entirely in Appendix A). Furthermore, this demodulation code provides a foundation upon which CPM signal reception can be built.

The first step for analyzing CPM over fiber is to reduce the vast parameter space to a finite set that will possibly achieve the desired behavior. This study is particularly interested in identifying CPM schemes capable of competing with QPSK, namely

providing the same SE and similar or better minimum distance in a filtered environment. To achieve this goal, the existing CPM minimum distance computation algorithm was modified to incorporate the influence of a channel filter when calculating the result. After doing so, it was found that quaternary CPFSK schemes with modulation index $0.25 \leq h \leq 0.4$ are most appropriate for the 100 Gb/s fiber application.

CPM receiver design depends upon the signal parameters; two architectures were developed for this effort, and both include all demodulation functions from the QPSK demodulator with modifications where necessary. The first, block-CPM architecture applies to signals that pass through the channel filters without significant loss. This architecture expands upon a previously existing design [42] that is compatible only with binary schemes, and leads to efficient processing. However, the linear equalizer is unable to achieve the full minimum distance potential of strongly filtered schemes as the out-of-band information lost through the channel filter cannot be recovered in such a receiver. For such CPM formats, the second architecture includes a DFE which generates a model of the channel for the feedback path. The transmitted signal is recreated within the DFE and recycled through the feedback taps to project its influence on the next decision. With these receivers, simulations for comparing CPM to other available formats are possible.

A variety of simulations indicated that the constant intensity per subcarrier of the CPM format is helpful for mitigating nonlinear phase modulation, however the relationship is indirect. The simulations uncovered that photonic interference between channels along the same SOP gives rise to vast swings in instantaneous power regardless of modulation format. The instantaneous power peak *levels* are modulation format dependent and are proportional to the peak power among each individual subcarrier.

Also, fiber dispersion plays a role in dissolving and reforming of peaks as they propagate through the fiber. Though CPM is an interesting modulation format, the simulations and experiments described herein indicate that its use in high speed fiber is limited to low SE systems (for min. distance gain over alternative formats) or highly nonlinear, low dispersion fibers (DSF, NZDF, SSMF @ 1310 nm, etc.).

Within the context of this dissertation, the following list identifies original contribution to the broader research community:

- Analysis of the use of CPM and OQPSK in fiber [65, 67, 68, 69, 72, 73, 74]
- A novel external modulator structure for CPM transmission [72]
- Demodulator code for 100G simulation and experimental testbed for QPSK, OQPSK, and CPM
- Extending the block CPM receiver design from [42] to cover M-ary full response formats [65, 73]
- Development of a DFE receiver for M-ary CPM [65]
- The use of blind asynchronous CMA for polarization demultiplexing [52], enabling a variety of modulation formats and sample rates
- Extension of the fast sequential minimum distance computation algorithm to operate over filtered CPM signals [submitted]
- QPSK format analysis for Ethernet scaling to 400 Gb/s and 1 Tb/s [71]

The contribution of demodulator code influenced the development of the following additional novel concepts, though full credit is due to the other researchers involved in the efforts:

- Nonlinear crosstalk penalty for 100 Gb/s QPSK systems [33, 34]

- Scaling studies: optimal dispersion maps and development of the NLT scaling parameter [30, 32, 75]
- Attaining absolute matching for BER vs. OSNR in experiments and simulations to within 0.5 dB [66]
- Super-channel receiver with active interference cancellation [submitted]

The full list of publications resulting from this work as primary author comprises eight conference papers and one journal article to date, with secondary contributions to six additional papers and articles. Three additional conference papers and one journal article are awaiting decisions. Two GTRC invention disclosures resulted; "External Optical Modulator Architecture for Producing Continuous Phase Modulation Signals", and "Super Receiver for Coherent Optical System".

7.2 Topics for Further Research

Moving forward form this effort, there remain several interesting studies available to enhance performance of 100 Gb/s systems and scale to higher speeds. As a fundamental building block, the use of ultra-large area fiber (ULAF) is being strongly considered as the media of choice for future phase-modulated networks. ULAF is beneficial since dispersion can be fully compensated digitally, and the larger area corresponds with a lower nonlinear coefficient. Beyond testing ULAF in the 100G Consortium's testbed, there are a number of interesting research opportunities related to this work. Among these are a multi-band optical transmitter architecture, an assessment of OFDM, star-QAM, and square QAM to increase SE and scale above 100 Gb/s, and a super-channel transceiver with active interference cancellation.

7.2.1 Multi-band optical transmitter

Coherent optical OFDM has been reported in recent years with a number of successful demonstrations [23, 24]. In some instances, subcarrier generation was performed by modulating the transmit laser source with a square wave (or approximation thereof) at some multiple of the subcarrier spacing frequency. In doing so, the transmitter is broken up into several sub-bands, each covering multiple subcarriers. Orthogonality is preserved based on the premise that all bands share the same carrier LO and band spacing clock. Each band has a corresponding set of modulators for I and Q components, and X and Y polarizations, and each modulator is driven by DACs that bear the subcarrier modulation tones. (Note that the experiments used a single set of modulators to cover the entire OFDM band rather than attempting to make each band independent). The benefit of breaking the modulation up into several smaller bands is that it enables the use of lower speed (and hence lower cost and power) electronic components to modulate a vast bandwidth.

Besides OFDM signal generation, this multi-band transmitter architecture can be leveraged to generate a single-carrier transmission. After all, OFDM signal generation allows for any arbitrary modulation scheme on a per-subcarrier basis, sending the set of subcarriers through an IFFT to generate the corresponding time-domain complex signal. An arbitrary waveform in the time domain could be generated by first creating the time domain signal, then passing it through an FFT followed by an IFFT. In fact, the IFFT could be split into N separate IFFTs, all fed by an FFT that is N times larger than each IFFT, provided that the IFFTs are frequency shifted from each other. This is the operating principle behind the multi-band transmitter, whose block diagram is shown in

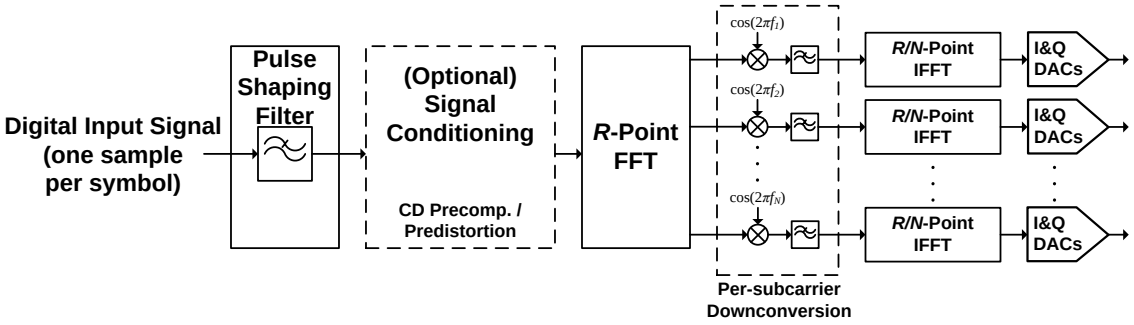


Fig. 7.1. Block diagram of multi-band transmitter structure [74]

Fig. 7.1 [74].

Besides OFDM, the multi-band transmitter may be used to economically generate CPM with several lower-speed DACs, higher-order QAM, or for arbitrary waveform generation. One challenge for this structure are to assure phase alignment of each modulated optical carrier, which could be assisted by some optical feedback path. This will limit the fidelity (i.e. effective number of bits) of the output signal. Also, it is unclear whether this method will solve any significant problems, or if it would be better to simply concentrate on making the DAC electronics faster. On the other hand, this could be a building block for a next-generation DAC, where the signal is generated in multiple banks and combined optically prior to emission (or conversion back to the electrical domain). It would also enable transmitter-side DSP for signal conditioning, such as self-biasing to compensate for low MZM extinction ratio, pulse shaping, and pre-distortion.

7.2.2 QAM and OFDM

Another interesting topic is the determination of the modulation format to scale past QPSK. There are several advantages of OFDM for moving forward, the primary being that the transmitter and receiver can both be broken up into an arbitrary number of bands of arbitrary bandwidth, thus dividing the problem of scaling down to a manageable

size. Furthermore, an OFDM system allows an arbitrary choice of per-subcarrier modulation type, which can be defined based on operating conditions. Therefore SE (and data rate) can be increased for high available OSNR or reduced for low OSNR. On the other hand, OFDM requires significant DSP resources at the transmitter which is a significant step up in complexity from QPSK. OFDM suffers from the well-known PAPR issue, which perhaps should be applied to describe nonlinear impairments in DWDM systems generally. It may be useful, therefore, to incorporate some of the PAPR reduction strategies already employed by OFDM systems (e.g. [25]) to mitigate peak formation at the transmitter.

Another alternative is QAM, whether the conventional square QAM or star QAM. The potential benefit of star QAM is that it would be more resilient to phase noise than square QAM, especially with increasing order, *M*. Star QAM has lower minimum distance in an AWGN environment, so a study to quantify performance of these formats is potentially useful. Also, QAM may be slightly easier to implement on the transmitter than OFDM, requiring a simpler DAC, or alternatively no DACs with careful optical signal combination [76].

In either case, scaling via increase in SE will continue to be an active area of interest for years to come. Feasibility and performance will be the main concerns, as will compatibility to past and future formats due to the high initial investment cost. To accommodate this concern, the development of a *universal transceiver* will be useful to cover everything from QPSK and OQPSK forward, including OFDM if possible. Fortunately, the polarization-diverse coherent optical receiver is a valuable building block that paves the way for a format-independent transceiver.

7.2.3 Super-channel receiver

Another future research goal is currently under research by the 100G consortium takes a different path to scaling than QAM and OFDM. With the availability of QPSK modulators and coherent optical receivers, it may be more cost-effective to scale up to 400 Gb/s or 1 Tb/s by packing several QPSK channels as close to each other as possible. To combat interference as the guard bands are dissolved, the concept of a super-channel receiver was proposed, which actively demodulates multiple channels in conjunction. Early results indicate that performance can be maintained while packing channels at a baud-rate separation.

APPENDIX C

100G TESTBED POLMUX DQPSK

DEMODULATOR CODE

This appendix contains the current revision of the Matlab code used in simulations and experiments for polarization multiplexed, differentially encoded QPSK. The code is presented in courier new font and colored to mimic its appearance in the Matlab editor.

A.1 Simulation Header

```
% This section is not run in a live simulation
% If the save_opt parameter is set in RSoft, the Rx_signal raw variable
% is saved in a specified file.  In a stand-alone mode (running demod
% on the file) set this to "if true"
if false % set these in the module parameters
    load('c:\file.mat');
    % RSoft signal inputs:
    XI = real(Rx_signal(1).Values);
    XQ = imag(Rx_signal(1).Values);
    YI = real(Rx_signal(2).Values);
    YQ = imag(Rx_signal(2).Values);

    % RSoft module parameters:
    poles = 2;
    ovs_rate = 25;
    decimation = 17;
    ADC_bits = 5; % can use fractional values
    CMA_tap_len = 7;
    CMA_mu = 0.0003;
    eq_mu = 0.001;
    t_rec_bl_len = 1000;
    c_rec_bl_len = 20;
    alpha = 0.8; % forgetting factor for phase
    delayY = 1250;
    diff_encode = 0;
    D = 0;
    debug = 1; % show plots
    save_opt = 0; % save Rx_signal data to .mat file
```

```
    init = 1; % first value for file name
    step_size = 1; % increment value for file name
    use_fde = 1; %1 for FDE, 0 for TDE
end
% this is required since RSoft will not modulate Y for the first x psec
% due to the DGD module for delaying and decorrelating X and Y
delayY_samps = delayY/1000*28*ovs_rate; % change 28 to baud rate if
necc.

% decimation
if decimation>13 % help suggests to break this up smaller
    initial_factors = factor(decimation);
    % first consolidate prime factors
    final_factors = ones(size(initial_factors));
    j=1;
    next_factor=1;
    for k=1:length(initial_factors)
        if next_factor*initial_factors(k)>13
            final_factors(j)=next_factor;
            next_factor=initial_factors(k);
            j=j+1;
        else
            next_factor=next_factor*initial_factors(k);
        end
    end
    final_factors(j)=next_factor;
    % now all the factors are < 13
    XIdec = double(XI(2+delayY_samps:end));
    XQdec = double(XQ(2+delayY_samps:end));
    YIdec = double(YI(2+delayY_samps:end));
    YQdec = double(YQ(2+delayY_samps:end));
    for k=1:length(final_factors)
        if final_factors(k)>1
            XIdec = decimate(XIdec,final_factors(k));
            XQdec = decimate(XQdec,final_factors(k));
            YIdec = decimate(YIdec,final_factors(k));
            YQdec = decimate(YQdec,final_factors(k));
        end
    end
else
    XIdec = decimate(double(XI(2+delayY_samps:end)),decimation);
    XQdec = decimate(double(XQ(2+delayY_samps:end)),decimation);
    YIdec = decimate(double(YI(2+delayY_samps:end)),decimation);
    YQdec = decimate(double(YQ(2+delayY_samps:end)),decimation);
end

% ADC quantization simulation
if (ADC_bits==-1) % no quantization
    XIquant = XIdec;
    XQquant = XQdec;
    YIquant = YIdec;
    YQquant = YQdec;
else % ADC_bits quantization
    range = max(max(abs([XIdec,YIdec,XQdec,YQdec])));
    step = range/(2^(ADC_bits-1)-1);
    XIquant = round(XIdec/step)*step;
```

```
    XQquant = round(XQdec/step)*step;
    YIquant = round(YIdec/step)*step;
    YQquant = round(YQdec/step)*step;
end
```

A.2 Demodulation Body

```
% CD compensation
h = fir1(63,0.14);
H = fft(h,128);
if D~=0
    %cd comp filter
    % first perform any necessary CD comp
    filt_len = 4096; % (max) selected length
    D_conv = -D*(1e-12/(1e-9)); % ps/nm-km => s/m^2
    lambda = 3e8/194e12; % m
    f0 = 28e9*ovs_rate/decimation; % Hz (Sampling Rate)

    % first define CD compensation filter in Freq Domain
    Hcd = zeros(2*filt_len,1);
    freqs = zeros(2*filt_len,1);
    %initialization: set Hcd according to all-pass transfer function
    for k=1:2*filt_len
        if k<=filt_len+1
            freq = (k-1)/(2*filt_len)*f0;
        else
            freq = -(2*filt_len-k+1)/(2*filt_len)*f0;
        end
        freqs(k) = freq;
        Hcd(k) = exp(1i*pi*D_conv*lambda^2*freq^2/3e8);
    end
    hcd_comp = ifft(Hcd);

    % compute Lh (99.99% power of taps in h[n])
    pwr = sum(abs(hcd_comp(1:filt_len)).^2);
    impulse_len = 0;
    tolerance = 0.0001;
    for k=1:filt_len
        if sum(abs(hcd_comp(k:filt_len)).^2)<tolerance*pwr
            impulse_len=2*k;
            break
        end
    end
    Lh = 2^ceil(log2(impulse_len));

    % TDE coefficients are symmetric and complex-valued
    hcd_comp_trunc = hcd_comp(Lh/2+1:-1:1);

    % FDE filter coefficients:
    Hcd = fft([hcd_comp_trunc(1:Lh/2+1);hcd_comp_trunc(Lh/2:-1:1); ...
              zeros(Lh-1,1)]);

    % TDE filter coefficients
```

```
    hcd_comp = [hcd_comp_trunc(1:Lh/2+1);hcd_comp_trunc(Lh/2:-1:1)];

    if use_fde % FDE
        len_samps = length(calSamps.Values(1,:));
        fftblocks = ceil(len_samps/Lh);
        Ex = zeros(1,(fftblocks+1)*Lh);
        Ey = zeros(1,(fftblocks+1)*Lh);

        for m=1:fftblocks
            if len_samps<m*Lh
                tmp=len_samps-(m-1)*Lh;
            else
                tmp=Lh;
            end

       % Hcd already "padded" to 2Lh above, pad X and Y inputs here
            Xblock = zeros(2*Lh,1);
            Yblock = zeros(2*Lh,1);
            Xblock(1:tmp) = calSamps.Values(1,1+(m-1)*Lh: ...
                            min(len_samps,m*Lh)).';
            Yblock(1:tmp) = calSamps.Values(2,1+(m-1)*Lh: ...
                            min(len_samps,m*Lh)).';

            Ex(1+(m-1)*Lh:(m+1)*Lh) = Ex(1+(m-1)*Lh:(m+1)*Lh) +...
                ifft(fft(Xblock).*Hcd).';
            Ey(1+(m-1)*Lh:(m+1)*Lh) = Ey(1+(m-1)*Lh:(m+1)*Lh) +...
                ifft(fft(Yblock).*Hcd).';
        end
        Tx = conv(calSamps.Values(1,:),hcd_comp);
        Xsamps = Ex(1+Lh/2:len_samps+Lh/2);
        Ysamps = Ey(1+Lh/2:len_samps+Lh/2);
    else
        Xsamps = conv(hcd_comp,calSamps.Values(1,:));
        Xsamps = Xsamps(1+Lh/2:len_samps+Lh/2);
        Ysamps = conv(hcd_comp,calSamps.Values(2,:));
        Ysamps = Ysamps(1+Lh/2:len_samps+Lh/2);
    end

else
    Xsamps = XIquant + 1i*XQquant;
    Ysamps = YIquant + 1i*YQquant;
end

% Remove DC Offset
sps = ovs_rate/decimation;

mean_p = (sqrt(mean(abs(Xsamps).^2))+sqrt(mean(abs(Ysamps).^2)))/2;
Xsamps = Xsamps/mean_p;
Ysamps = Ysamps/mean_p;

% CMA equalizer
% initialize to zeros except XX center tap
w_xx = zeros(1,CMA_tap_len);
w_xy = zeros(1,CMA_tap_len);
w_yx = zeros(1,CMA_tap_len);
```

```
w_yy = zeros(1,CMA_tap_len);
w_xx(ceil(CMA_tap_len/2)) = 1;

% train once to estimate SOP...
trunc = min([10000,length(Xsamps),length(Ysamps)]); % initialize CMA
                                                    % with <= 10000 samples
[out_x, out_y, w_xx, w_xy, w_yx, w_yy, e_x, e_y, hx, hy] = ...
    xpol_eq_func(Xsamps(1:trunc), Ysamps(1:trunc), 0, 1, poles, ...
                 CMA_mu*3, 50, w_xx, w_xy, w_yx, w_yy);

% ... then use estimate to extract x & y components
[out_x, out_y, w_xx, w_xy, w_yx, w_yy, e_x, e_y, hx, hy] = ...
    xpol_eq_func(Xsamps, Ysamps, 0, 1, 0, CMA_mu, 50, ...
                 w_xx, w_xy, w_yx, w_yy);


% Timing Recovery & Interpolation
if interp_meth<2 % for cubic or polynomial:
    % Digital Filter & Square recovery with non-integer oversampling
    % rate
    interp_offset = 3;
    interp_len = length(out_x)-interp_offset+1;
    x = abs(out_x(interp_offset:interp_offset+interp_len-1)).^2;
    y = exp(1i*2*pi*(0:interp_len-1)*decimation/ovs_rate);
    tmp = y*x;
    m = interp_offset;
    mu_x = 1/(2*pi)*angle(tmp)*ovs_rate/decimation;
    if mu_x<0
        mu_x = mu_x+ovs_rate/decimation;
    end

    if poles==2
        interp_len = length(out_y)-interp_offset+1;
        x = abs(out_y(interp_offset:interp_offset+interp_len-1)).^2;
        y = exp(1i*2*pi*(0:interp_len-1)*decimation/ovs_rate);
        tmp = y*x;
        m = interp_offset;
        mu_y = 1/(2*pi)*angle(tmp)*ovs_rate/decimation;
        if mu_y<0
            mu_y = mu_y+ovs_rate/decimation;
        end
    end
end

if interp_meth==0
    % cubic interpolation:
    m  = interp_offset;
    mu = mu_x;
    k = 1;
    while m+1<length(out_x)
        trec_outX(k) = out_x(m-2)*(1/6)*(-mu^3+  mu) + ...
                       out_x(m-1)*(1/2)*( mu^3+  mu^2-2*mu) + ...
                       out_x(m)  *(1/2)*(-mu^3-2*mu^2+  mu+ 2) + ...
                       out_x(m+1)*(1/6)*( mu^3+3*mu^2+2*mu);
        k = k+1;
        next = m+mu+0.5*(ovs_rate/decimation); % use 0.5 for 2
```

```
                                                    % samples/symbol
        m = floor(next);
        mu = next-m;
    end

    if poles==2
        m  = interp_offset;
        mu = mu_y;
        k = 1;
        while m+1<length(out_y)
            trec_outY(k)= out_y(m-2)*(1/6)*(-mu^3+  mu) + ...
                          out_y(m-1)*(1/2)*( mu^3+  mu^2-2*mu) + ...
                          out_y(m)  *(1/2)*(-mu^3-2*mu^2+  mu+ 2) + ...
                          out_y(m+1)*(1/6)*( mu^3+3*mu^2+2*mu);
            k = k+1;
            next = m+mu+0.5*(ovs_rate/decimation); % use 0.5 for 2
                                                   % samples/symbol
            m = floor(next);
            mu = next-m;
        end
    end
end

if interp_meth==1
    % polynomial interpolation:
    C0 = [ 0.00000,  0.00000,  1.00000,  0.00000,  0.00000,  0.00000];
    C1 = [ 0.10951, -0.64994, -0.20510,  1.01184, -0.33578,  0.07683];
    C2 = [-0.14218,  0.96411, -1.57796,  0.77122,  0.02162, -0.04416];
    C3 = [ 0.03267, -0.31416,  0.78306, -0.78306,  0.31416, -0.03267];

    m  = interp_offset;
    mu = mu_x;
    k = 1;
    while m+3<length(out_x)
        y0 = C0*out_x(m-2:m+3);
        y1 = C1*out_x(m-2:m+3);
        y2 = C2*out_x(m-2:m+3);
        y3 = C3*out_x(m-2:m+3);
        trec_outX(k) = y0+mu*(y1+mu*(y2+mu*y3));
        k = k+1;
        next = m+mu+0.5*(ovs_rate/decimation); % use 0.5 for 2
                                               % samples/symbol
        m = floor(next);
        mu = next-m;
    end

    if poles==2
        m  = interp_offset;
        mu = mu_y;
        k = 1;
        while m+3<length(out_y)
            y0 = C0*out_y(m-2:m+3);
            y1 = C1*out_y(m-2:m+3);
            y2 = C2*out_y(m-2:m+3);
            y3 = C3*out_y(m-2:m+3);
            trec_outY(k) = y0+mu*(y1+mu*(y2+mu*y3));
```

```
            k = k+1;
            next = m+mu+0.5*(ovs_rate/decimation); % use 0.5 for 2
                                                   % samples/symbol
            m = floor(next);
            mu = next-m;
        end
    end
end

% Timing Recovery via Upsample/FIR/Downsample
if interp_meth==2
    for k=0:num_t_bl-1
        x = abs(sym_x(k*t_rec_bl_len*ovs_rate+1: ...
                (k+1)*t_rec_bl_len*ovs_rate)).^2;
        y = exp(1i*2*pi*(0:t_rec_bl_len*ovs_rate-1)/ovs_rate);
        tmp = y*x;
        tau(k+1) = 1/(2*pi)*angle(tmp)*ovs_rate-1;
        if tau(k+1)<0
            tau(k+1) = tau(k+1)+ovs_rate;
        end

        for m=1:t_rec_bl_len % use 2 samples per symbol
            trec_outX((k*t_rec_bl_len+m)*2-1) = ...
                sym_x((k*t_rec_bl_len+m-1)*ovs_rate+1+ ...
                floor(tau(k+1)))*(1-tau(k+1)+floor(tau(k+1)));
            trec_outX((k*t_rec_bl_len+m)*2-1) = ...
                trec_outX((k*t_rec_bl_len+m)*2-1) +...
                sym_x((k*t_rec_bl_len+m-1)*ovs_rate+1+...
                ceil(tau(k+1)))*(1-ceil(tau(k+1))+tau(k+1));
            trec_outX((k*t_rec_bl_len+m)*2)   = ...
                sym_x((k*t_rec_bl_len+m-1)*ovs_rate+1+...
                floor(tau(k+1)+ovs_rate/2))*(1-(tau(k+1)+...
                ovs_rate/2)+floor(tau(k+1)+ovs_rate/2));
            trec_outX((k*t_rec_bl_len+m)*2)   = ...
                trec_outX((k*t_rec_bl_len+m)*2) + ...
                sym_x((k*t_rec_bl_len+m-1)*ovs_rate+1+...
                ceil(tau(k+1)+ovs_rate/2))*(1-ceil(tau(k+1)+...
                ovs_rate/2)+(tau(k+1)+ovs_rate/2));
        end
    end

    if poles==2
        for k=0:num_t_bl-1
            x = abs(sym_y(k*t_rec_bl_len*ovs_rate+1:(k+1)*...
                t_rec_bl_len*ovs_rate)).^2;
            y = exp(1i*2*pi*(0:t_rec_bl_len*ovs_rate-1)/ovs_rate);
            tmp = y*x;
            tau(k+1) = 1/(2*pi)*angle(tmp)*ovs_rate-1;
            if tau(k+1)<0
                tau(k+1) = tau(k+1)+ovs_rate;
            end

            for m=1:t_rec_bl_len % use 2 samples per symbol
                trec_outY((k*t_rec_bl_len+m)*2-1) = ...
                    sym_y((k*t_rec_bl_len+m-1)*ovs_rate+1+...
                    floor(tau(k+1)))*(1-tau(k+1)+floor(tau(k+1)));
```

```
                trec_outY((k*t_rec_bl_len+m)*2-1) = ...
                    trec_outY((k*t_rec_bl_len+m)*2-1) + ...
                    sym_y((k*t_rec_bl_len+m-1)*ovs_rate+1+...
                    ceil(tau(k+1)))*(1-ceil(tau(k+1))+tau(k+1));
                trec_outY((k*t_rec_bl_len+m)*2)   = ...
                    sym_y((k*t_rec_bl_len+m-1)*ovs_rate+1+...
                    floor(tau(k+1)+ovs_rate/2))*...
                    (1-(tau(k+1)+ovs_rate/2)+...
                    floor(tau(k+1)+ovs_rate/2));
                trec_outY((k*t_rec_bl_len+m)*2)   = ...
                    trec_outY((k*t_rec_bl_len+m)*2) + ...
                    sym_y((k*t_rec_bl_len+m-1)*ovs_rate+1+...
                    ceil(tau(k+1)+ovs_rate/2))*...
                    (1-ceil(tau(k+1)+ovs_rate/2)+...
                    (tau(k+1)+ovs_rate/2));
            end
        end
    end
end

% set up equalizer inputs (Ts/2 spacing)
eq_inx = zeros(1,length(trec_outX));
x_1sps = trec_outX(1:2:end);
eq_iny = zeros(1,length(trec_outY));
y_1sps = trec_outY(1:2:end);

% Frequency offset correction:
% use an odd number of taps
if mod(c_rec_bl_len,2)==0
    max_taps = c_rec_bl_len+1;
else
    max_taps = c_rec_bl_len;
end

phx4_fft = abs(fft(x_1sps.^4,freq_fft_len));
freq_ofs = (find(max(phx4_fft)==phx4_fft)-1);
if freq_ofs>(freq_fft_len/2)
    freq_ofs = freq_ofs-freq_fft_len;
end
freq_ofs = freq_ofs/(4*freq_fft_len);

runlen=length(x_1sps);

x_no_ofs=x_1sps.*exp(-1i*(0:runlen-1)*2*pi*freq_ofs);

eq_inx = trec_outX.*exp(-1i*(0:2*runlen-1)*pi*freq_ofs);

if poles==2
    phx4_fft = abs(fft(y_1sps.^4,freq_fft_len));
    freq_ofs = (find(max(phx4_fft)==phx4_fft)-1);
    if freq_ofs>(freq_fft_len/2)
        freq_ofs = freq_ofs-freq_fft_len;
    end
    freq_ofs = freq_ofs/(4*freq_fft_len);
```

```
    runlen=length(y_1sps);

    y_no_ofs=y_1sps.*exp(-1i*(0:runlen-1)*2*pi*freq_ofs);

    eq_iny = trec_outY.*exp(-1i*(0:2*runlen-1)*pi*freq_ofs);
end

% decision directed equalizer
if resetEQ==1
    % reset taps
    eq_qpskx = lineareq(15,lms(eq_mu*3),qammod([0:3],4)/sqrt(2),2);
    eq_qpskx.RefTap = 9;
    eq_qpskx.Weights = zeros(1,15);
    eq_qpskx.Weights(eq_qpskx.RefTap)=1;
    eq_qpskx.ResetBeforeFiltering = 0;
    eq_inx = eq_inx-mean(eq_inx);
    eq_inx = eq_inx(1:length(eq_inx)-mod(length(eq_inx),2));

    [eq_outx, out_d, e] = equalize(eq_qpskx,eq_inx);

    eq_qpskx.StepSize = eq_mu;
end

w_savex = eq_qpskx.Weights;
eq_inx = eq_inx(1:length(eq_inx)-mod(length(eq_inx),2));

[eq_outx, out_d, e] = equalize(eq_qpskx,eq_inx);

if poles==2
    if resetEQ==1
        eq_qpsky = lineareq(15,lms(eq_mu*3),qammod([0:3],4)/sqrt(2),2);
        eq_qpsky.RefTap = 9;
        eq_qpsky.Weights = zeros(1,15);
        eq_qpsky.Weights(eq_qpsky.RefTap)=1;
        eq_qpsky.ResetBeforeFiltering = 0;
        eq_iny = eq_iny-mean(eq_iny);
        eq_iny = eq_iny(1:length(eq_iny)-mod(length(eq_iny),2));
        [eq_outy, out_d, e] = equalize(eq_qpsky,eq_iny);
        eq_qpsky.StepSize = eq_mu;
    end

    w_savey = eq_qpsky.Weights;
    eq_iny = eq_iny(1:length(eq_iny)-mod(length(eq_iny),2));
    [eq_outy, out_d, e] = equalize(eq_qpsky,eq_iny);
 end

demod_symx = eq_outx(length(w_savex):end-length(w_savex));
demod_symx = demod_symx(:).';

if poles==2
    demod_symy = eq_outy(length(w_savey):end-length(w_savey));
    demod_symy = demod_symy(:).';
end
```

```
% Carrier Phase Estimation
% use an odd number of taps
if mod(c_rec_bl_len,2)==0
    max_taps = c_rec_bl_len+1;
else
    max_taps = c_rec_bl_len;
end

z=demod_symx.^4;
z=z./abs(z);
z = z(:);
c = zeros(1,2*max_taps+1);
%compute autocorrelation for MMSE filter
for i=1:2*max_taps+1
    if i==max_taps
        %debug = 1;
    end
    c(i) = (z(max_taps+1:length(demod_symx)-
max_taps)')*z(i:length(demod_symx)+i-2*max_taps-1);
end
%normalize center tap
c = c/abs(c(max_taps+1));
% add noise variance for coeff computation

OSNR_lin = 10^(osnr_appx/10);
gamma = OSNR_lin * 2*Bref * Tsym;
noise_var = 0;
for p=1:4
    noise_var = noise_var+nchoosek(4,p)^2*factorial(p)/(gamma^p);
end

c_n = c*(1+noise_var);
% generate autocorrelation matrix
acm = zeros(max_taps);
for i=0:max_taps-1
    acm(i+1,:) = c_n(max_taps+1-i:2*max_taps-i);
end

% compute MMSE filter coefficients
% acm * a = c, where acm's top row is c(0) to c(max_taps+1)
% acm's left column is c(0) to c(-max_taps-1)
% and c is a column vector of c(-center_tap+1:center_tap-1)
center_tap = ceil(max_taps/2);
c_rhs = c_n(max_taps+1-center_tap+1:max_taps+1+center_tap-1).';
c_rhs((max_taps+1)/2) = 1;
a = (acm^-1)*(c_rhs);

% Compute Weiner filter coefficients
if use_mmse_ph_est == 3
    D = center_tap-1;
    M = 4;
    alpha = 0.8;
    num(1) = (1-alpha)*alpha^D;
    for k = 1:D
        num(k+1) = (1-alpha)^2*alpha^(D-k);
    end
```

```
    den = zeros(1,length(num));
    den(1) = 1;
    den(2) = alpha;

    h_weiner = filter(num,den,[1; zeros(2*D,1)]);
end

if use_mmse_ph_est == 2
    a = ones(size(a))/length(a);
elseif use_mmse_ph_est == 3
    a = h_weiner;
end

z=demod_symx.^4;
z=-z./abs(z);
z = z(:);
car_phs = zeros(1,length(z));
car_phs = unwrap(angle(conv(a,z)))/4;

if (use_mmse_ph_est~=3)
    demod_symx = demod_symx.*exp(-1i*car_phs(center_tap:end-
(center_tap-1))).';
    demod_symx = demod_symx(center_tap:end-center_tap);
else
    demod_symx = demod_symx.*exp(-1i*car_phs(center_tap:end-
(center_tap-1))).';
    demod_symx = demod_symx(center_tap:end-center_tap);
end

car_phsx = car_phs; % for debug
car_phsx_taps = a; % for debug

if poles==2
    if mod(c_rec_bl_len,2)==0
        max_taps = c_rec_bl_len+1;
    else
        max_taps = c_rec_bl_len;
    end

    z=demod_symy.^4;
    z=z./abs(z);
    z = z(:);
    c = zeros(1,2*max_taps+1);
    %compute autocorrelation
    for i=1:2*max_taps+1
        if i==max_taps
            %debug = 1;
        end
        c(i) = (z(max_taps+1:length(demod_symx)-
max_taps)')*z(i:length(demod_symx)+i-2*max_taps-1);
    end
    %normalize center tap
    c = c/abs(c(max_taps+1));
    % add noise variance for coeff computation
```

```
    OSNR_lin = 10^(osnr_appx/10);
    gamma = OSNR_lin * 2*Bref * Tsym;
    noise_var = 0;
    for p=1:4
        noise_var = noise_var+nchoosek(4,p)^2*factorial(p)/(gamma^p);
    end

    c_n = c*(1+noise_var);
    % generate autocorrelation matrix
    acm = zeros(max_taps);
    for i=0:max_taps-1
        acm(i+1,:) = c_n(max_taps+1-i:2*max_taps-i);
    end

    % compute MMSE filter coefficients
    % acm * a = c, where acm's top row is c(0) to c(max_taps+1)
    % acm's left column is c(0) to c(-max_taps-1)
    % and c is a column vector of c(-center_tap+1:center_tap-1)
    center_tap = ceil(max_taps/2);
    c_rhs = c_n(max_taps+1-center_tap+1:max_taps+1+center_tap-1).';
    c_rhs((max_taps+1)/2) = 1;
    a = (acm^-1)*(c_rhs);

    if use_mmse_ph_est == 2
        a = ones(size(a))/length(a);
    elseif use_mmse_ph_est == 3
        a = h_weiner;
    end

    z=demod_symy.^4;
    z=-z./abs(z);
    z = z(:);
    car_phs = zeros(1,length(z));
    car_phs = unwrap(angle(conv(a,z)))/4;

    if use_mmse_ph_est ~= 3
        demod_symy = demod_symy.*exp(-1i*car_phs(center_tap:...
            end-(center_tap-1))).';
        demod_symy = demod_symy(center_tap:end-center_tap);
    else
        demod_symy = demod_symy.*exp(-1i*car_phs(center_tap:...
            end-(center_tap-1))).';
        demod_symy = demod_symy(center_tap:end-center_tap);
    end

    car_phsy = car_phs; % for debug
    car_phsy_taps = a; % for debug
end

% remove DC offset (if any) from DDE output
demod_symx = demod_symx - mean(demod_symx);
if poles==2
    demod_symy = demod_symy - mean(demod_symy);
end
```

```
% get bits
bitsXRE = real(demod_symx)>0;
bitsXIM = imag(demod_symx)>0;

errsRE = zeros(1,2);
errsIM = zeros(1,2);
symsRE = zeros(1,2);
symsIM = zeros(1,2);
clear diffXRE diffXIM;
[diffXRE, diffXIM] = diff_decode(bitsXRE, bitsXIM, 0,0);

[junk, errvecRE, first_matchRE, polRE] = prbs_match(bitsXRE);
[junk, errvecIM, first_matchIM, polIM] = prbs_match(bitsXIM);
first_match = max(first_matchRE,first_matchIM);
if first_match>0
    [seqXRE] = prbs15_gen(length(bitsXRE)+1-first_match, polRE, ...
        bitsXRE(first_match:first_match+14));
    [seqXIM] = prbs15_gen(length(bitsXIM)+1-first_match, polIM, ...
        bitsXIM(first_match:first_match+14));
    if diff_encode==0
        errsRE(1) = sum(xor(seqXRE(2:end),bitsXRE(first_match+1:end)));
        errsIM(1) = sum(xor(seqXIM(2:end),bitsXIM(first_match+1:end)));
        symsRE(1)=length(seqXRE)-1;
        symsIM(1)=length(seqXIM)-1;
    else
        [diffSeqXRE, diffSeqXIM] = diff_decode(seqXRE, seqXIM, 0,0);
        errsRE(1) = sum(xor(diffSeqXRE(2:end),diffXRE(first_match+1:...
            first_match+length(diffSeqXRE)-1)));
        errsIM(1) = sum(xor(diffSeqXIM(2:end),diffXIM(first_match+1:...
            first_match+length(diffSeqXIM)-1)));
        symsRE(1)=length(diffSeqXRE)-1;
        symsIM(1)=length(diffSeqXIM)-1;
    end
end

if poles==2
    % get bits for Y pol
    bitsYRE = real(demod_symy)>0;
    bitsYIM = imag(demod_symy)>0;

    clear diffYRE diffYIM;
    [diffYRE, diffYIM] = diff_decode(bitsYRE, bitsYIM, 0,0);

    [junk, errvecRE, first_matchRE, polRE] = prbs_match(bitsYRE);
    [junk, errvecIM, first_matchIM, polIM] = prbs_match(bitsYIM);
    first_match = max(first_matchRE,first_matchIM);
    if first_match>0
        [seqYRE] = prbs15_gen(length(bitsYRE)+1-first_match, polRE,...
            bitsYRE(first_match:first_match+14));
        [seqYIM] = prbs15_gen(length(bitsYIM)+1-first_match, polIM,...
            bitsYIM(first_match:first_match+14));
        if diff_encode==0
            errsRE(2) = sum(xor(seqYRE(2:end),bitsYRE(first_match+1:...
                end)));
            errsIM(2) = sum(xor(seqYIM(2:end),bitsYIM(first_match+1:...
                end)));
```

```
            symsRE(2)=length(seqYRE)-1;
            symsIM(2)=length(seqYIM)-1;
        else
            [diffSeqYRE, diffSeqYIM] = diff_decode(seqYRE,seqYIM,0,0);
            errsRE(2)= sum(xor(diffSeqYRE(2:end),diffYRE(first_match...
                +1:first_match+length(diffSeqYRE)-1)));
            errsIM(2)= sum(xor(diffSeqYIM(2:end),diffYIM(first_match...
                +1:first_match+length(diffSeqYIM)-1)));
            symsRE(2)=length(diffSeqYRE)-1;
            symsIM(2)=length(diffSeqYIM)-1;

        end
    end

else
    errsRE(2)=0;
    errsIM(2)=0;
    symsRE(2)=0;
    symsIM(2)=0;
end;

errsX = (sum(errsRE(1))+sum(errsIM(1)));
errsY = (sum(errsRE(2))+sum(errsIM(2)));
totErrs = errsX + errsY;
BERx = (sum(errsRE(1))+sum(errsIM(1)))/(sum(symsRE(1))+sum(symsIM(1)));
BERy = (sum(errsRE(2))+sum(errsIM(2)))/(sum(symsRE(2))+sum(symsIM(2)));
totSyms = sum(symsRE)+sum(symsIM);
BER = (sum(errsRE)+sum(errsIM))/(sum(symsRE)+sum(symsIM));
```

A.3 Supporting Functions

A.3.1 diff_decode

```
% Differentially Decode a DQPSK symbol stream
function [outI, outQ] = diff_decode(inI, inQ, initI, initQ)

seq = ([initI; inI(:)]-0.5) + 1i*([initQ; inQ(:)]-0.5);
diff = mod((angle(seq(2:end))-angle(seq(1:end-1)))*2/pi,4);
diffM = diff>1;
diffL = mod(diff,2);

outI = diffM';
outQ = xor(diffM,diffL)';

end
```

A.3.2 prbs15_gen

```
% Generate PRBS-15 pattern with specified length, polarity, and
% seed
function [output] = prbs15_gen(len, pol, seed)
```

```
    output = (zeros(1,len));
    if length(seed)~=15
        seed=ones(1,15);
    end
    if sum(seed)==0
        seed=ones(1,15);
    end
    if pol==-1
        output(1:15) = not(seed);
    else
        output(1:15) = seed;
    end
    for j=16:len
        output(j) = xor(output(j-15),output(j-14));
    end
    if pol==-1
        output = not(output);
    end
end
```

A.3.3 prbs_match

```
% Match the incoming sequence with the PRBS-15 sequence, reporting
% the number of errors, error locations, first match location, and
% polarity of matched vector
function [errs, errvec, first_match, pol] = prbs_match(in_seq)

    % polynomial 1+X^14+X^15
    consec_match = 0;
    errvecp = zeros(size(in_seq));
    first_matchp = 0;
    max_consec_match = 0;

    % try non-inverted:
    phits = 0;
    for j=16:length(in_seq)
        if in_seq(j)~=xor(in_seq(j-15),in_seq(j-14))
            phits = phits+1;
            errvecp(j) = 1;
            consec_match = 0;
        else
            if consec_match>15 && first_matchp==0
                first_matchp = j-31;
            end
            consec_match = consec_match+1;
            max_consec_match = max(consec_match,max_consec_match);
        end
    end
    if first_matchp>0
        prbsv = prbs15_gen(length(in_seq)+1-first_matchp, 1, ...
            in_seq(first_matchp:first_matchp+14));
        phits = sum(xor(in_seq(first_matchp:end),prbsv'));
    else
        phits = length(in_seq);
    end
```

```
    % try inverted:
    consec_match = 0;
    errvecn = zeros(size(in_seq));
    first_matchn = 0;
    inv_seq = not(in_seq);
    nhits = 0;
    for j=16:length(inv_seq)
        if inv_seq(j)~=xor(inv_seq(j-15),inv_seq(j-14))
            nhits = nhits+1;
            errvecn(j) = 1;
            consec_match = 0;
        else
            if consec_match>15 && first_matchn==0
                first_matchn = j-31;
            end
            consec_match = consec_match+1;
            max_consec_match = max(consec_match,max_consec_match);
        end
    end
    if first_matchn>0
        prbsv = prbs15_gen(length(inv_seq)+1-first_matchn, -1, ...
            in_seq(first_matchn:first_matchn+14));
        nhits = sum(xor(in_seq(first_matchn:end),prbsv'));
    else
        nhits = length(inv_seq);
    end

    if (nhits<phits)
        pol = -1;
        errs = nhits;
        errvec = errvecn;
        first_match = first_matchn;
    else
        pol = 1;
        errs = phits;
        errvec = errvecp;
        first_match = first_matchp;
    end
end
```

A.3.4 xpol_eq_func

```
% Cross polarized equalization function; can operate in CMA (blind)
% or DDE modes, inputs unequalized signals across two SOPs, can
% optionally perform carrier recovery, outputs equalized signal,
% updated tap weights, error stats
function [out_x, out_y, wo_xx, wo_xy, wo_yx, wo_yy, eo_x, eo_y, ...
    high_errx, high_erry] = xpol_eq_func(uneq_x, uneq_y, car_rec,...
    blind_mode, pol_mode, step, block, wi_xx, wi_xy, wi_yx, wi_yy)

% symbol-spaced equalizer
% initialize weights
wo_xx = wi_xx; % weights;
wo_xy = wi_xy; % weights;
```

```
wo_yx = wi_yx; % weights;
wo_yy = wi_yy; % weights;
eq_x = zeros(length(uneq_x)+length(wi_xx),1);
eq_y = zeros(length(uneq_y)+length(wi_yy),1);
out_x = zeros(length(uneq_x)+length(wi_xx),1);
out_y = zeros(length(uneq_y)+length(wi_yy),1);
eo_x = zeros(length(uneq_x)+length(wi_xx),1);
eo_y = zeros(length(uneq_y)+length(wi_xx),1);
high_errx = zeros(length(uneq_x)+length(wi_xx),1);
high_erry = zeros(length(uneq_y)+length(wi_xx),1);
mu = step; %0.01;
blind_eq = blind_mode; %1;
e_x=0;
e_y=0;
carr_x_est = 0;
carr_y_est = 0;

for j = 1:length(uneq_x)+length(wo_xx)-1
    % perform filtering
    for k = 1:length(wo_xx)
        if j-k+1<1 || j-k+1>length(uneq_x)
            continue;
        end;
        eq_x(j) = eq_x(j) + uneq_x(j-k+1)*(wo_xx(k)) + ...
            uneq_y(j-k+1)*(wo_xy(k));
        eq_y(j) = eq_y(j) + uneq_x(j-k+1)*(wo_yx(k)) + ...
            uneq_y(j-k+1)*(wo_yy(k));
    end;
    out_x(j) = eq_x(j)*exp(-i*angle(carr_x_est)/4);
    out_y(j) = eq_y(j)*exp(-i*angle(carr_y_est)/4);

    % carrier recovery
    if car_rec
        if mod(j,block)==0
            carr_x_est=0;
            carr_y_est=0;
            for k=j-block+1:j
                carr_x_est = carr_x_est + eq_x(k)^4;
                carr_y_est = carr_y_est + eq_y(k)^4;
            end
        end
    end

    if j<length(wo_xx)
        %assume zero error for first length(wo_xx) symbols
        e_x = 0;
        e_y = 0;

    elseif blind_mode==1 % CMA
        % for QPSK, take E[|x(n)|^4]/E|x(n)|^2] as 1/1 = 1
        e_x = (1-abs(eq_x(j))^2)*eq_x(j);
        e_y = (1-abs(eq_y(j))^2)*eq_y(j);
        if pol_mode==1
            e_y = -out_y(j);
        end
    else
```

```
        % once converged, go to LMS
        dec_x = (sign(real(out_x(j))) +
1i*sign(imag(out_x(j))))/sqrt(2);
        dec_y = (sign(real(out_y(j))) +
1i*sign(imag(out_y(j))))/sqrt(2);
        e_x = dec_x-out_x(j);
        e_y = dec_y-out_y(j);
    end

    for k = 1:length(wo_xx)
        if j-k+1<1 || j-k+1>length(uneq_x)
            continue;
        end;
        wo_xx(k) = wo_xx(k) + mu*e_x*conj(uneq_x(j-k+1));
        wo_xy(k) = wo_xy(k) + mu*e_x*conj(uneq_y(j-k+1));
        wo_yx(k) = wo_yx(k) + mu*e_y*conj(uneq_x(j-k+1));
        wo_yy(k) = wo_yy(k) + mu*e_y*conj(uneq_y(j-k+1));
    end;
    eo_x(j) = e_x;
    eo_y(j) = e_y;
    if abs(e_x).^2>1
        high_errx(j)=1;
    end
    if abs(e_y).^2>1
        high_erry(j)=1;
    end
    if pol_mode>0
        % use physical channel meaning:
        % [w_xx  w_xy] [x] = [h]
        % [w_yx  w_yy] [y] = [v]
        % [u   v ]
        % [-v* u*]
        u = (wo_xx + conj(wo_yy(end:-1:1)))/2;
        v = (wo_xy + -conj(wo_yx(end:-1:1)))/2;
        if length(wo_xx)==1
            scale = sqrt(det([u,v;-conj(v),conj(u)]));
        else
            scale = 1;
        end
        wo_xx = u/scale;
        wo_xy = v/scale;
        wo_yx = -conj(v(end:-1:1))/scale;
        wo_yy = conj(u(end:-1:1))/scale;
    end

end;

scale = sqrt(det(sqrt([sum(abs(wo_xx).^2), sum(abs(wo_xy).^2); ...
    sum(abs(wo_yx).^2), sum(abs(wo_yy).^2)])));

center_x = mod(find(max(abs([wo_xx(:); wo_xy(:)]).^2)==...
                        (abs([wo_xx(:); wo_xy(:)]).^2)),length(wo_xx));
center_y = mod(find(max(abs([wo_yx(:); wo_yy(:)]).^2)==...
                        (abs([wo_yx(:); wo_yy(:)]).^2)),length(wo_yy));
%set to 1 for 1-tap filter
center_x = max(center_x,1);
```

```
center_y = max(center_y,1);

out_x = out_x(center_x:length(uneq_x)+center_x-1);
out_y = out_y(center_y:length(uneq_y)+center_y-1);
eo_x = eo_x(center_x:length(uneq_x)+center_x-1);
eo_y = eo_y(center_y:length(uneq_y)+center_y-1);
high_errx = high_errx(center_x:length(uneq_x)+center_x-1);
high_erry = high_erry(center_y:length(uneq_y)+center_y-1);
```

APPENDIX B

BLOCK-MODE CPM RECEIVER DESIGN

This appendix contains the full description of the block CPM receiver design, as published in Sec. V of the upcoming JLT article. [65]

B.1 Block-based CPM Receiver Architecture

Optimal reception of CPM signals generally requires a bank of filters matched to all possible sequences of symbols over the designed observation interval, after which the maximal likelihood symbol sequence must be determined [36]. As such, implementation of a CPM receiver is notably more complex than receivers for other phase modulation schemes like QPSK, and similar to those in the maximum a posteriori sequence detection of non-zero ISI schemes [77]. Furthermore, unlike QPSK there are no known CPM receiver architectures compatible with optical differential direct detection. Coherent reception is required to achieve adequate performance in comparison to other available formats.

A coherent optical receiver (Fig. 2.4) outputs photocurrents proportional to the fields along each of two arbitrary linear polarizations. These photocurrents must first be sampled while locked to the symbol rate (synchronous sampling) or with a free-running clock above twice the bandwidth of the I and Q signals (asynchronous sampling) then all subsequent processing is handled by DSP. Note that in asynchronous sampled systems, a frequency $<2/T_S$ may be used as long as the signal power aliased from the second Nyquist

band is negligible. The proposed CPM receiver architecture is based on the linear decomposition of the CPM signal, a technique first proposed for binary formats by Laurent [38] and later extended to *M*-ary schemes [39]. Through linear decomposition, the CPM signal is described as a superposition of complex-valued pulse amplitude modulation (PAM) waveforms, which enables the use of linear demodulation techniques such as channel equalization. The PAM pulses themselves are real-valued, finite duration, and determined by the CPM parameters h, M, L, and the window shape. For *M*-ary CPM schemes, the PAM pulses are called Laurent functions (LFs). These PAM pulses are multiplied by data dependent complex-valued coefficients are called pseudo-symbols (PSs), which exhibit a temporal correlation to maintain the continuity and restricted progression of phase states. LF and PS generation is discussed in detail in Sec. V.B. Note that use of linear decomposition at the receiver does not require changes to the CPM transmitter.

A block-based transmission approach to CPM [42, 78] enables efficient frequency domain equalization (FDE) of the channel and LFs for binary schemes. The system architecture draws parallels to OFDM, with the clear difference in the underlying signal (namely OFDM performs subcarrier modulation via IFFT operation). The block structure of the CPM signal imposes several restrictions on the transmitter, including insertion of a cyclic prefix and the "intrafix". The cyclic prefix transforms the channel equalizer into a circulant matrix for efficient ISI cancellation, as it does for OFDM. The duration of the cyclic prefix (N_P symbols) must exceed the channel delay spread, but kept small since it is overhead. The intrafix is unique to CPM, as it ensures that the initial phase state of the cyclic prefix matches the initial phase state of the final N_P symbols of the block. As a

result of the intrafix, the desired correlation is achieved, i.e. $s(t) = s(t+N_P T_S)$ during the cyclic prefix.

The receiver (Fig. 5.10) processes blocks of CPM symbols (length N_T) sampled at $T_S/2$ intervals. Initially the receiver applies a filter to compensate for the majority of residual CD. This is performed immediately due to three factors; i) to reduce the possibly large delay spread CD imposes (thus reducing the required cyclic prefix length), ii) the impact of CD is well known and readily compensated [79], and iii) the operation can be cascaded with subsequent channel equalizers due to its all-pass response. After CD compensation the polarization modes are demultiplexed via constant modulus algorithm, with each of the transmitted modes sent to its own synchronization module, where symbol timing and carrier phase are recovered. Next the data blocks are stripped of their cyclic prefix, and converted to the frequency domain (FD) via N-point FFT (where $N=N_T - N_P$). The channel equalizer is applied to the blocks to further eliminate intersymbol interference. Then the Laurent functions, which are computed by the underlying linear decomposition, are equalized and processed by N-point IFFT to return to the time domain (TD). Branch metrics of the symbol sequences are generated and undergo maximum likelihood sequence detection in a Viterbi decoder according to the phase trellis of the CPM scheme. Finally, the intrafix is discarded and the data symbols are passed on to FEC decoding.

The preferred order of processing was specified to reduce implementation complexity and transmission overhead, though other arrangements of the receiver modules are possible. Additionally, the CD compensation filter may be eliminated if an optical method is used, or can be implemented via FDE for reduced complexity. Note

that the intrinsic correlation of the cyclic prefix with the final symbols of the block can aid in symbol timing recovery, a technique common in OFDM systems [45].

B.2 Reception of M-ary CPM Full-Response Formats

As explained in Sec. IV, CPFSK schemes are of particular interest for SE > 1 b/s/Hz since these full response schemes lead to the least complex implementation without excessive penalty compared to alternative CPM formats. The following description is valid for any *M*-ary full-response system, based on the binary receiver in [42]. The receiver assumes that blocks of length N_T symbols are transmitted, with a cyclic prefix of N_P symbols. The modulation index of the system is *h*, which is assumed to be a ratio of coprime integers *m*/*p*. An intrafix of *K* symbols precedes the final N_P symbols of the block, where *K*=ceil[(*p*-1)/(*M*-1)]. The definition of a CPM signal may be expressed in its linear decomposition format as

$$s(t) = \sum_{k=0}^{M-2} \sum_{n} a_{k,n} g_k (t - nT_S) \tag{1}$$

where the LFs $g_k(t)$ are determined by the following relationships to $q(t)$ and *h*:

$$g_k(t) = \prod_{\ell=0}^{P-1} u^{(\ell)}(t + e_{o,\ell}^{(k)} T_S) \quad k = 0,\ldots M-2 \tag{2}$$

$$u^{(\ell)}(t) = \begin{cases} \sin\left(2^{\ell+1} h \pi q(t)\right) / \sin\left(2^{\ell} h \pi\right) & 0 \le t \le T_S \\ u^{(\ell)}(2T_S - t) & T_S \le t \le 2T_S \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Note that P is defined as $\log_2(M)$, and the coefficients $e_{o,\ell}^{(k)}$ take on all the *M*-1 permutations of 0's and 1's except for all 1's. The PSs $a_{k,n}$ are related to the Laurent decomposition Pseudo-Coefficients (PCs) b_n by:

$$a_{k,n} = \prod_{\ell=0}^{P-1} b_{n-e_{o,\ell}^{(k)}}^{(\ell)} \qquad k = 0,\ldots M-2 \tag{4}$$

$$b_n^{(\ell)} = \exp\left\{ j 2^{\ell} h \pi \sum_{m=-\infty}^{n} \gamma_{m,\ell} \right\} \tag{5}$$

where $\gamma_{m,\ell}$ are the binary symbol values (±1) per bit position ℓ in the P-bit representation of the m^{th} symbol.

Under the assumption that most of the CPM signal energy lies within the frequency range $1/T_S$ of the carrier, such that $T_S/2$ sampling is at the Nyquist rate, the transmitted signal may be discretized as

$$s_n^i = \sum_{k=0}^{M-2} \sum_m a_{k,m} g_{k,n-m}^i \tag{6}$$

$$g_{k,n}^i = g_k(t)\Big|_{t=T_S(n+i/2)} \tag{7}$$

The CPM signal passes through a linear channel with impulse response $h(t)=\psi_{tx}(t)*\psi_{ch}(t)*\psi_{rx}(t)$, with contributions from the transmit filter, cascaded ROADM filters, and receive filter. Noise is added, and is assumed to be AWGN prior to coloration by the ROADM filters and receive filter. The received baseband signal is therefore

$$r(t) = \sum_n s_n h(t - nT_S/2) + \tilde{\eta}(t) \tag{8}$$

After sampling at $T_S/2$ intervals, the received signal has two polyphase components i=0,1:

$$r_n^i = \sum_m s_m^0 h_{n-m}^i + \sum_m s_m^1 h_{n-m-1+i}^{(1-i)} + \tilde{\eta}_n^i \tag{9}$$

where

$$h_k^i = h_k(t)\Big|_{t=T_S(k+i/2)} \tag{10}$$

$$\tilde{\eta}_k^i = \tilde{\eta}_k(t)\Big|_{t=T_S(k+i/2)}. \tag{11}$$

The linear channel is expressed in N_TxN_T matrix form as

$$\left[\underline{\underline{h}}_0^{i(\ell)}\right] = h_{n-m}^{i(\ell)}, \quad i = 0,1 \tag{12}$$

$$\left[\underline{\underline{h}}_0^{1'(\ell)}\right] = h_{n-m-1}^{1(\ell)} \tag{13}$$

$$\left[\underline{\underline{h}}_1^{i(\ell)}\right] = h_{N_T+n-m}^{i(\ell-1)}, \quad i = 0,1 \tag{14}$$

$$\left[\underline{\underline{h}}_1^{1'(\ell)}\right] = h_{N_T+n-m-1}^{1(\ell-1)}. \tag{15}$$

The cyclic prefix in this scheme is inserted to ensure no inter-block interference, as long as the delay spread of the channel (after CD compensation) is less than $N_P \cdot T_S$, as well as to facilitate FDE. The cyclic prefix insertion ($\underline{\underline{T}}_{CP}$) and deletion ($\underline{\underline{R}}_{CP}$) are modeled in matrix form as

$$\underline{\underline{T}}_{CP} = \left[\underline{\underline{0}}_{N_P \times (N-N_P)}, \underline{\underline{I}}_{N_P}; \underline{\underline{I}}_N\right] \tag{16}$$

$$\underline{\underline{R}}_{CP} = \left[\underline{\underline{0}}_{N \times N_P}, \underline{\underline{I}}_N\right]. \tag{17}$$

Thus the polyphase received signal is represented in vector-matrix form by

$$\begin{aligned} r &= \begin{bmatrix} \underline{\underline{R}}_{CP}\,\underline{\underline{h}}^{0} & \underline{\underline{R}}_{CP}\,\underline{\underline{h}}^{1'} \\ \underline{\underline{R}}_{CP}\,\underline{\underline{h}}^{1} & \underline{\underline{R}}_{CP}\,\underline{\underline{h}}^{0} \end{bmatrix} \cdot \\ &\qquad \begin{bmatrix} g_0^0 \underline{\underline{T}}_{CP} & g_1^0 \underline{\underline{T}}_{CP} & \cdots & g_{P-1}^0 \underline{\underline{T}}_{CP} \\ g_0^1 \underline{\underline{T}}_{CP} & g_1^1 \underline{\underline{T}}_{CP} & \cdots & g_{P-1}^1 \underline{\underline{T}}_{CP} \end{bmatrix} \underline{a} + \underline{\tilde{\eta}} \end{aligned} \tag{18}$$

$$= \underline{\underline{h}}\,\underline{\underline{g}}\,\underline{a} + \underline{\tilde{\eta}} \tag{19}$$

To reduce implementation complexity, channel equalization is performed in the FD. Since all the channel and LF submatrices are circulant, they can be transformed by the *N*x*N* discrete Fourier transform (DFT) matrix $\underline{\underline{F}}_N$ into diagonal submatrices. So (19) in the FD is written as

$$\underline{R} = \underline{\underline{H}}\,\underline{\underline{G}}\,\underline{A} + \underline{N} \tag{20}$$

The FD representation of the transmitted signal be defined separately as

$$\underline{S} = \underline{\underline{G}}\,\underline{A} \tag{21}$$

Using the MMSE equalizer proposed in [80], the minimum mean square estimate of the transmitted signal is given as

$$\underline{\hat{S}} = \underline{\underline{C}}_{MMSE}\ \underline{R} \tag{22}$$

where

$$\underline{\underline{C}}_{MMSE} = \left[\left(\mathcal{E}\left\{ \underline{S}\,\underline{S}^{H} \right\} \right)^{-1} + \frac{1}{\sigma_n^2} \underline{\underline{H}}^{H}\,\underline{\underline{H}} \right]^{-1} \frac{1}{\sigma_n^2} \underline{\underline{H}}^{H} \tag{23}$$

The final step in demodulation is to pass the equalized signal through a matched filter bank and Viterbi decoder. The filter bank is matched to the LFs of the scheme, which operates in the FD, then the IFFT is used to return to the TD for processing the symbols in the TD. Symbol decisions are subject to maximizing the metric

$$\Lambda = \underline{\tilde{S}}^{H}\,\underline{\hat{S}} \tag{24}$$

$$= \underline{\tilde{a}}^{H}\,\underline{\underline{F}}_{N,P}^{H}\,\underline{\underline{G}}^{H}\,\underline{\hat{S}} \tag{25}$$

$$= \underline{\tilde{a}}^{H}\,\underline{z} \tag{26}$$

$$\underline{z} = \underline{\underline{F}}_{N,P}^{H}\,\underline{\underline{G}}^{H}\,\underline{\hat{S}} \tag{27}$$

The vector $\underline{z}$ acts as the output of a bank of M-1 filters matched to the LFs. The branch metric unit must calculate each of the possible metrics over all trellis branches i=1,…,I as

$$\lambda_n^i = \mathrm{Re}\left(\sum_{k=0}^{M-2} z_{k,n}\,\tilde{a}_k^{i\,*} \right) \tag{28}$$

B.3 Simplification for M-ary CPM Full-Response Formats

The receiver process given above for extension to full-response *M*-ary formats can be simplified by making a few observations. First, it is noted that the LF $g_0(t)$ is $2T_S$ in duration, all other LFs ($g_1(t)$ through $g_{M-2}(t)$) are T_S long, all start at $g_k(0)=0$ and end at $g_0(t)=0|_{t=2Ts}$ or $g_k>0(t)=0|_{t=Ts\ (k>0)}$. This results from the fact that all underlying $u_k(t)$ signals are zero at $t=0$ and $t=2T_S$, and unity at $t=T_S$, and are furthermore symmetric about $t=T_S$. So in a two-phase implementation, the only nonzero tap in the 0-phase is at $g_0(t=T_S)=1$, whereas the 1-phase has nonzero taps at $g_0(t=T_S/2)$, $g_0(t=3T_S/2)$, and $g_k>0(t=T_S/2)$. It is also noted that the symmetry in the underlying $u_k(t)$ signals (i.e. $u_k(T_S-t)= u_k(T_S+t)$) provides that all the nonzero 1-phase taps are equal, and take the value

$$\xi = g_0\left(t = \frac{T_S}{2}\right) = \frac{\sin(\pi h/2)}{\sin(M\pi h/2)} \tag{29}$$

The PSs, $a_{k,n}$ from (4), comprise a set of $2p$ phase states (assuming m is odd in the relation $h=m/p$) at time intervals $t=nT_S$, where n is an arbitrary integer. (Note that only p phase states exist for even m). With this in mind, and given the prior consideration to the two-phase sampled form of $g_k(t)$, consider a 4-ary CPFSK scheme in the context of (6) and (7):

$$s_n^0 = \sum_m a_{0,m}\, g_{0,n-m}^0 \Big|_{g_{0,n-m}^0 = \delta[n-m]} \tag{30}$$

$$s_n^0 = b_n^0\, b_n^1 \tag{31}$$

$$s_n^1 = \left. \begin{array}{l} \sum_m a_{0,m} g^1_{0,n-m} + \sum_m a_{0,m-1} g^1_{0,n-m+1} + \\ \sum_m a_{1,m} g^1_{1,n-m} + \sum_m a_{2,m} g^1_{2,n-m} \end{array} \right| \begin{array}{l} g^1_{0,n-m} = \xi\delta[n-m]; \\ g^1_{1,n-m} = \xi\delta[n-m]; \\ g^1_{2,n-m} = \xi\delta[n-m] \end{array} \tag{32}$$

$$\begin{aligned} s_n^1 &= \xi\left(b_n^0 b_n^1 + b_{n+1}^0 b_n^1 + b_n^0 b_{n+1}^1 + b_{n+1}^0 b_{n+1}^1\right) \\ &= \exp\left\{ j\left[\arg(s_n^0) + \arg(s_n^1)\right]/2 \right\} \end{aligned} \tag{33}$$

Naturally, this result agrees with the nonlinear baseband representation of the CPFSK signal; i.e. combining (5), (31), and (33) yields

$$s_n^0 = \exp\left\{ jh\pi \sum_{m=-\infty}^{n} \gamma_{m,0} + j2h\pi \sum_{m=-\infty}^{n} \gamma_{m,1} \right\} \tag{34}$$

$$s_n^1 = \exp\left\{ jh\pi \sum_{m=-\infty}^{n} \gamma_{m,0} + j2h\pi \sum_{m=-\infty}^{n} \gamma_{m,1} + j\frac{h}{2}\pi\gamma_{n+1,0} + jh\pi\gamma_{n+1,1} \right\} \tag{35}$$

Note that the expression for indicates that the phase state of the signal on the 0-phase sampling points $s(t=nT_S)$ are restricted to the $2p$ phase states 0, $\pi\cdot 1/p$, ... $\pi\cdot(2p\text{-}1)/p$. Note that the symbol values for $\gamma_{n,0}$ and $\gamma_{n,1}$ are ±1, so only p of the $2p$ states (0, $\pi\cdot 2/p$, ... $\pi\cdot(2p\text{-}2)/p$) are reachable on the 0-phase even symbols, and the other p are reachable on the odd symbols ($\pi\cdot 1/p$, $\pi\cdot 3/p$, ... $\pi\cdot(2p\text{-}1)/p$). Although p phase states are reachable, only M phase states are allowed by transitions from the previous symbol's phase state. For high SE CPFSK schemes, $M \le p$ (m=1) is typical. The 1-phase samples (s_n^1) can take on any of the $2p$ phase states $\pi\cdot 1/(2p)$, $\pi\cdot 3/(2p)$, ... $\pi\cdot(2p\text{-}1)/(2p)$.

An example of the CPFSK phase trellis for M=4, h=1/4 is shown in Fig. B.1. In the proposed receiver architecture, instead of using an LF equalizer to reduce the signal to an estimate for the PSs, the receiver takes the equalized FD representation of the

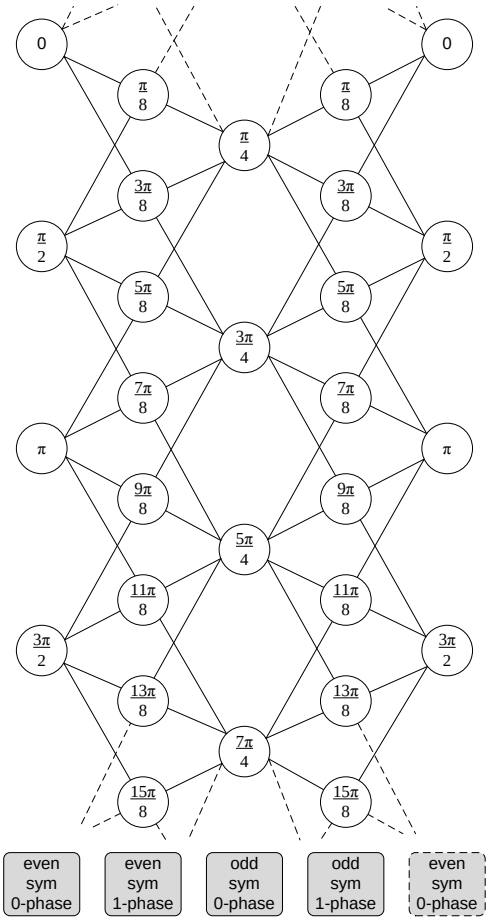


Fig. B.1. CPFSK (*M*=4, *h*=1/4) phase state trellis for 2-phase processing

received signal, and converts back to the time domain for branch metric calculations. In other words, (24) is modified as

$$\begin{aligned} \Lambda &= \underline{\tilde{S}}^{H}\,\underline{\hat{S}} \\ \lambda &= \underline{\tilde{s}}^{H}\,\underline{\underline{F}}_{N,P}^{H}\,\underline{\hat{S}} \end{aligned} \tag{36}$$

and the branch metric calculations are taken over all possible states i=0,1,...I as

$$\lambda_n^{i} = \mathrm{Re}\left(\hat{s}_n^{0i}\tilde{s}_n^{0i*} + \hat{s}_n^{1i}\tilde{s}_n^{1i*}\right) \tag{37}$$

Under this simplification, the receiver drops the complexity of LF equalization, and directly computes branch metrics on the signal's phase transitions rather than calculating metrics over the abstract PSs. This simplification may be used for any full-response CPM signal, and can be applied to any scheme with the "separable phase"

property as described in [81] (since these schemes collapse to full-response equivalent schemes). It is noted that the branch metric unit may take advantage of the symmetry of the possible states, and only needs to perform complex multiplications on half the possible states while the other half are derived by simply negating the calculated ones. This can be applied to any format for the 1-phase branch metric calculations, and for formats with even p on the 0-phase calculations.

Finally, we note that the proposed architecture is most suitable for those formats which fit well within the ROADM bandwidth. For higher modulation indices, the intended CPM signal is significantly distorted by the elimination of sidelobe power. In these cases, the frequency domain equalizer is eliminated (as is the cyclic prefix), and is replaced by a decision feedback equalizer (DFE) in the branch metric unit. The DFE is capable of restoring the intended envelope of the previously detected symbols for the branch metric unit, and may be combined with soft decision feedback to achieve the minimum distance available to lower SE formats (e.g. $h \geq 1/4$ 4-ary CPFSK).

APPENDIX C

CPM MODULATOR AND DEMODULATOR CODE

This appendix contains the current revision of the Matlab code for modulating and demodulating CPM signals in RSoft OptSim simulation environment. The code is presented in courier new font and colored to mimic its appearance in the Matlab editor.

C.1 Modulator

The modulator is capable of transmitting either QPSK (input parameter "format" set to -1) or CPFSK ($h=1/3$ when "format" is set to 0, otherwise h="format"). The output signals can either drive V_I and V_Q control voltage signals on a nested MZM QM, or can be connected to a single MZM AM and PM for better signal modeling on side channels (this is selected by the "seed" parameter, 0 for main channel, non-zero for side channels). Note that "seed" also provides the initial state for the random number generators.

```
M = 4;

% for debug: fill these in:
if 0
    format=0;
    ovs_rate = 25;
    Vpi = 2.5;
    V0 = 0;
    num_samples = 8000;
    seed= 1;
end
num_symbols = ceil(num_samples/ovs_rate)+1;
tx_filt = fir1(ovs_rate,3/ovs_rate,'low');

if format<0
    tx_mod = zeros(num_symbols*ovs_rate+17,1);

    % use "temporal" raised cosine pulse to model electronic signal
    % swing behavior
```

```
    rcpulse = zeros(1,ovs_rate+17);
    rcpulse(1:17) = cos(pi:-pi/16:0)*0.5+0.5;
    rcpulse(17:ovs_rate+2) = 1;
    rcpulse(ovs_rate+1:ovs_rate+17) = cos(0:pi/16:pi)*0.5+0.5;
    data_seq = (randint(num_symbols,1,M,seed)-1.5)*2;
    for i=1:num_symbols
        tx_mod((i-1)*ovs_rate+1:i*ovs_rate+17) = tx_mod((i-1)*...
            ovs_rate+1:i*ovs_rate+17)+ ...
            rcpulse.'* ((data_seq(i)>0)-0.5)*2;
        tx_mod((i-1)*ovs_rate+1:i*ovs_rate+17) = tx_mod((i-1)*...
            ovs_rate+1:i*ovs_rate+17)+ ...
            1i*rcpulse.'*((abs(data_seq(i))>2)-0.5)*2;
    end

    pulse_carve = cos(pi/4*(1+cos([0:num_samples-1]*2/ovs_rate*pi)));
    % on a scale of 0 = Vpi and 1 = Von:
    pulse_carve = 1-(1+cos([0:num_samples-1]*2/ovs_rate*pi))/2;
    pulse_carve = pulse_carve(:);

    if seed==0 %use i and q
        tx_mod = tx_mod(34:num_samples+33).*pulse_carve;

        Vi = (real(tx_mod)+1)*Vpi;
        Vq = (imag(tx_mod)+1)*Vpi;
    else % use A and theta, and give it a fractional bitslot offset
        offset = round(mod(seed,5)/5*ovs_rate);
        Vi = (angle(tx_mod(34+offset:num_samples+offset+33)))*Vpi/pi;

        Vq = pulse_carve.*(sqrt(abs(tx_mod(34+offset:...
            num_samples+offset+33).^2)/2));
        Vq = Vpi*(1-Vq);
        %Eq = cos(Vq*pi/(2*Vpi));
    end
else

    %R = 28;
    % initial phase is 0 degrees
    step = 0;
    curr = 0;
    next = 0;

    if format==0
        h=1/3;
    elseif format>1
        h=format-1;
    else
        h=format;
    end

    % generate data blocks
    data_seq = (randint(num_symbols,1,M,seed)-1.5)*2;
    tx_seq = zeros(num_symbols*ovs_rate,1);
    if seed==0
        jstart = 50;
    else
```

```
    jstart = 1;
end

if seed==0
    tmp = ovs_rate;
    ovs_rate=2;
    tx_seq = zeros(num_symbols*ovs_rate,1);
end

for j=jstart:num_symbols
    if mod(ovs_rate,2)==0
        tx_seq((j-0.5)*ovs_rate+1:(j+0.5)*ovs_rate) = ...
           tx_seq((j-1)*ovs_rate+1)+...
           data_seq(j)*[0.5*ones(ovs_rate/2,1);ones(ovs_rate/2,1)];
    else
        tmp2 = floor(ovs_rate/2);
        tx_seq((j-0.5)*ovs_rate+0.5) = ...
            tx_seq((j-1)*ovs_rate+1)+data_seq(j)*0.25;
        tx_seq((j-0.5)*ovs_rate+1.5:(j+0.5)*ovs_rate-0.5) = ...
            tx_seq((j-1)*ovs_rate+1)+data_seq(j)*...
            [0.5*ones(tmp2,1);ones(tmp2,1)];
    end
end

if seed==0
    ovs_rate = tmp;
end

if seed==0 % normal MZM for center channel
    fact_string = factor(round(h*9*16*25*7*11));
    rot = h/4;
    for i=5:length(fact_string)
        if fact_string(i)==2
            rot = rot/2;
        else
            break
        end
    end

    tx_seq = exp(1i*tx_seq*h*pi).*exp(1i*pi*rot)*0.5;

    if mod(ovs_rate,2)==1
        rcpulse2 = zeros(1,ovs_rate+21);
        rcpulse2(1:21) = cos(pi:-pi/20:0)*0.5+0.5;
        rcpulse2(21:ovs_rate+2) = 1;
        rcpulse2(ovs_rate+1:ovs_rate+21) = ...
            cos(0:pi/20:pi)*0.5+0.5;
    else
        rcpulse2 = zeros(1,ovs_rate/2+11);
        rcpulse2(1:11) = cos(pi:-pi/10:0)*0.5+0.5;
        rcpulse2(11:ovs_rate/2+2) = 1;
        rcpulse2(ovs_rate/2+1:ovs_rate/2+11) = ...
            cos(0:pi/10:pi)*0.5+0.5;
    end
```

```
        drive_i = (1-acos(real(tx_seq))/pi)*2*Vpi;
        drive_q = (1-acos(imag(tx_seq))/pi)*2*Vpi;

        if mod(ovs_rate,2)==0
            tx_mod = zeros(num_samples+ovs_rate*3,1);
            for i=1:length(tx_seq)
                tx_mod((i-1)*ovs_rate/2+1:i*ovs_rate/2+11) = ...
                    tx_mod((i-1)*ovs_rate/2+1:i*ovs_rate/2+11)+   ...
                    rcpulse2.'*drive_i(i);
                tx_mod((i-1)*ovs_rate/2+1:i*ovs_rate/2+11) = ...
                    tx_mod((i-1)*ovs_rate/2+1:i*ovs_rate/2+11)+ ...
                    1i*rcpulse2.'*drive_q(i);
            end
            tx_mod = tx_mod(ovs_rate:end);
        else
            tx_mod = zeros(2*(num_samples+ovs_rate*3),1);
            for i=1:length(tx_seq)
                tx_mod((i-1)*ovs_rate+1:i*ovs_rate+21) = ...
                    tx_mod((i-1)*ovs_rate+1:i*ovs_rate+21)+  ...
                    rcpulse2.'*drive_i(i);
                tx_mod((i-1)*ovs_rate+1:i*ovs_rate+21) = ...
                    tx_mod((i-1)*ovs_rate+1:i*ovs_rate+21)+ ...
                    1i*rcpulse2.'*drive_q(i);
            end
            tx_mod = tx_mod(ovs_rate:2:end);
        end
        Vi = real(tx_mod(1:num_samples));
        Vq = imag(tx_mod(1:num_samples));
        Ei = cos(Vi*pi/(2*Vpi));
        Eq = cos(Vq*pi/(2*Vpi));
        plot(Ei,Eq);
    else % for side-channels, use just Vpi
        Vi = zeros(num_samples,1);
        offset = round(mod(seed,5)/5*ovs_rate);

        Vi(:,1) = (tx_seq(1+offset:num_samples+offset)*h)*Vpi;
        % Vq is connected to amplitude modulator
        Vq = zeros(num_samples,1);
    end
end;
```

C.2 Demodulator

The demodulator matches the modulator for the “format” and “seed” parameter definitions. The demodulator includes a function for initial training of the DFE weight taps, cpm_ch_est. Note that phase estimation is not included in this revision as the simulation laser linewidth was set to zero to isolate fiber nonlinearities as the sole source of phase noise.

```
limit = length(Rx_signal(1).Values);

Xsamps = Rx_signal(1).Values(1:limit);
Ysamps = Rx_signal(2).Values(1:limit);
mean_p = (sqrt(mean(abs(Xsamps).^2))+sqrt(mean(abs(Ysamps).^2)))/2;
Xsamps = Xsamps/mean_p;
Ysamps = Ysamps/mean_p;

wxx = [0,0,0,1,0,0,0];
wxy = [0,0,0,0,0,0,0];
wyx = [0,0,0,0,0,0,0];
wyy = [0,0,0,0,0,0,0];

trn_len = 50000;
trn_end = min(trn_len,length(Xsamps));
[junkx,junky, wxx,wxy,wyx,wyy, junkex,junkey, junkhx,junkhy] = ...
    xpol_eq_func(Xsamps(1:trn_end), ...
    Ysamps(1:trn_end), ...
    0, 1, 2, 0.001, 50, wxx, wxy, wyx, wyy);
[Xpol,Ypol, wxx,wxy,wyx,wyy, junkex,junkey, junkhx,junkhy] = ...
    xpol_eq_func(Xsamps(1:trn_end), ...
    Ysamps(1:trn_end), ...
    0, 1, 2, 0.001, 50, wxx, wxy, wyx, wyy);

% no explicit phase noise, so just use preamble for phase offset
delay = 33;
phase0 = angle(mean(Xpol(1:delay)))
rx_b2b = Xpol(delay:end).*exp(-1i*phase0);

num_symbols = ceil(49+length(rx_b2b)/2);
if format==0
    m=1;
    p=4;
else
    p=0;
    for k=1:32
        m=round(format*k);
        if abs(m-format*k)<0.001
            p=k;
            break;
        end
    end
end

if format==2/7 % just in case, avoid rounding error and match tx:
    format = 0.2857142857;
end
seed = 0;
M=4;
if format==0
    h=1/3;
elseif format>1
    h=format-1;
```

```
else
    h=format;
end

% generate data blocks
data_seq = (randint(num_symbols+5,1,M,seed)-1.5)*2;
tx_seq = zeros(num_symbols*2+10,1);
if seed==0
    jstart = 50;
else
    jstart = 1;
end
for j=jstart:num_symbols+5
    tx_seq((j-0.5)*2+1:(j+0.5)*2) = tx_seq((j-1)*2+1)+...
        data_seq(j)*[0.5;1];
end
tx_cpfsk = exp(1i*tx_seq(99:end)*h*pi);

tx_sig_offset = 229; % TX PWR = 2 dBm
tx_sig_offset = 1; % TX PWR = 0 dBm
tx_cpfsk = tx_cpfsk(tx_sig_offset:end);

%rx_b2b_x2 = resample(double(rx_b2b),2,1);
update_weights=1;

if update_weights

    trn_end=min(length(rx_b2b),50000);

    weights = zeros(3,13); % 7 taps, centered

    [out, weights(1,:), eo] = cpm_ch_est(rx_b2b(1:trn_end), ...
        tx_cpfsk(5:trn_end+4), 0.003, weights(1,:));
    [out, weights(2,:), eo] = cpm_ch_est(rx_b2b(1:trn_end), ...
        tx_cpfsk(3:trn_end+2), 0.003, weights(2,:));
    [out, weights(3,:), eo] = cpm_ch_est(rx_b2b(1:trn_end), ...
        tx_cpfsk(1:trn_end), 0.003, weights(3,:));
end

ovsrate = 2;
h = m/p;
serrs =0;
cmp_len=0;
for pol=1:poles

    if pol==1
        rx_sig = Xpol(delay:end).*exp(-1i*phase0);
    else
        rx_sig = Ypol(delay:end).*exp(-1i*phase0);
    end
    rx_sig = rx_sig(:).';
    tx_sig = tx_cpfsk;

    % storage:
    branch_metric = zeros(p,M);
```

```
path_metric = zeros(p,1);
survivor = zeros(p,traceback);

% initialize to exp(0), since we happen to know that the first
% state isexp(0)
path_metric(p) = 2;

% give the range of possible data (given traceback length)
dataset = -(M-1)*ones(traceback,M^traceback);
for k=1:M^traceback-1
   str = dec2bin(k,log2(M)*traceback);
   for l=1:traceback
       dataset(l,k+1)=bin2dec(str(l*2-1:l*2))*2-(M-1);
   end
end

% now build a table from possible states with dataset to produce
% possiblephases
phaseset = zeros(traceback*2,M^traceback);
phaseset(1,:) = dataset(1,:)*m/(2*p);
phaseset(2,:) = dataset(1,:)*m/p;
for k=2:traceback
    phaseset(k*2-1,:)=phaseset(k*2-2,:)+dataset(k,:)*m/(2*p);
    phaseset(k*2,:)  =phaseset(k*2-2,:)+dataset(k,:)*m/p;
end

rx_decided = zeros(1,floor(length(rx_sig)/2));
rx_decided(1:length(weights(1,:))-2) = ...
    % just say this is part of a frame preamble
    tx_sig(1:length(weights(1,:))-2);

pathset = zeros(traceback*2,M^traceback);
pathmet = 0;

delay_weights = 2*traceback-1;
transset = zeros(1,13+length(weights(1,:)));
filtset  = zeros(traceback,13);

xx = zeros(1,2+2*traceback);
for j=length(weights):2:length(rx_sig)-4
  % given:  at outset, we have most recent say 4 symbol
  % decisionsnow we need to project all combinations of
  % next (traceback) symbols

    curr_state = angle(rx_decided(j-2))/pi;
                % modulus makes this on the range 1 to 2*p

    pathset = exp(1i*pi*(curr_state+phaseset));
    transset(1:delay_weights+6) = ...
        rx_decided(j-8-delay_weights+1:j-2);
    minmet = 100;
    decision = 0;

     for k=1:M^traceback
        transset(delay_weights+7:delay_weights+7+...
```

```
                (2*traceback-1)) = pathset(:,k).';
            transset(delay_weights+7+(2*traceback):end) = ...
                pathset(end,k);

            if traceback==2
                 xx = conv(weights(2,1:9),transset(1:14));
                 xx = xx(9:14);
            end
            if traceback==3
                for l=1:6
                    xx(l) = weights(1,end:-1:1)*...
                        transset(l:l-1+length(weights(1,:))).';
                end
                for l=7:8
                    xx(l) = weights(2,end:-1:1)*...
                        transset(l-2:l-2-1+length(weights(1,:))).';
                end
            end
            pathmet = sqrt(sum(abs(xx-...
                rx_sig(j-4:j+2*traceback-3)).^2));

            if pathmet<minmet
                minmet = pathmet;
                decision = k;
                xd = xx;
            end
        end
        rx_decided(j-1:j)=pathset(1:2,decision);
    end

    tx_data = data_seq(50:end);

    rx_unwrap = unwrap(angle(rx_decided));
    rx_data = round((rx_unwrap(3:2:end)-rx_unwrap(1:2:end-2))*...
        p/(m*pi));

    rx_data_ofs = 1;
    tx_data_ofs = 1+floor(tx_sig_offset/2);
    cmp_len = cmp_len + min(length(tx_data)-tx_data_ofs,...
        length(rx_data)-rx_data_ofs);

    err_list = find(round(tx_data(tx_data_ofs:tx_data_ofs+...
        cmp_len-1))~=round(rx_data(rx_data_ofs:rx_data_ofs+...
        cmp_len-1).'));
    serrs = serrs + length(err_list);
end
SER = serrs/cmp_len
```

C.3 Supporting Functions

C.3.1 cpm_ch_est

```
function [out, wo, eo] = cpm_ch_est(received, expected, step, wi)

wo = wi; % weights;

%% init dedicion-directed CMA adaption

eq = zeros(length(expected)+length(wi),1);
out = zeros(length(expected)+length(wi),1);
eo = zeros(length(expected)+length(wi),1);
mu = step; %0.01;
e=0;

%% CMA w/ e_dd = |a_i|^2 - |s_i|^2
for j = 2:min(length(expected),length(received))-1

    % perform filtering
    for k = 1:length(wo)
        if j-k+1<1 || j-k+1>length(expected)
            continue;
        end;
        eq(j) = eq(j) + expected(j-k+1)*(wo(k));
    end;

    % error calculation and weight updates
    e = received(j)-eq(j);

    for k = 1:length(wo)
        if j-k+1<1 || j-k+1>length(expected)
            continue;
        end;
        wo(k) = wo(k) + mu*e*conj(expected(j-k+1));
    end;
    eo(j) = e;

    out(j) = eq(j);
end
```

REFERENCES

[1] A. Gnauck, R. Tkach, A. Chraplyvy, and T. Li, "High-capacity optical transmission systems," *J. Lightw. Technol.*, vol. 26, no. 9, pp. 1032-1045, May 2008.

[2] R. Ramaswami and K. Sivarajan, *Optical Networks,* 2nd ed. San Francisco: Morgan Kaufmann, 2002.

[3] *Ciena 5400 Family of Reconfigurable Switching Systems,* http://media.ciena.com/documents/ActivFlex_5410_Reconfigurable_Switching_System_DS.pdf

[4] A. Gnauck, G. Charlet, P. Tran, P. Winzer, C. Doerr, J. Centanni, E. Burrows, T. Kawanishi, T. Sakamoto, and K. Higuma, "25.6-Tb/s C+L-band transmission of polarization-multiplexed RZ-DQPSK signals," *OFC 2007*, PDP19.

[5] C. Fludger, T. Duthel, D. van den Borne, C. Schulien, E.-D. Schmidt, T. Wuth, J. Geyer, E. De Man, G.-D. Khoe, and H. de Waardt, "Coherent equalization and POLMUX-RZ-DQPSK for robust 100-GE transmission," *J. Lightw. Technol.*, vol. 26, no. 1, pp. 62-74, Jan. 2008.

[6] P. Winzer, G. Raybon, H. Song, A. Adamiecki, S. Corteselli, A. Gnauck, D. Fishman, C. Doerr, S. Chandrasekhar, L. Buhl, T. Xia, G. Wellbrock, W. Lee, B. Basch, T. Kawanishi, K. Higuma, and Y. Painchaud, "100-Gb/s DQPSK Transmission: From Laboratory Experiments to Field Trials," *J. Lightw. Technol.*, vol. 26, no. 20, pp. 3388-3402, Oct. 2008.

[7] H. Wernz, S. Bayer, B. E. Olsson, M. Camera, H. Griesser, and C. Fürst, " 112Gb/s PolMux RZ-DQPSK with fast polarization tracking based on interference control," *OFC 2009*, Mar. 2009.

[8] *100G Ultra Long Haul DWDM Framework Document,* http://www.oiforum.com/public/documents/OIF-FD-100G-DWDM-01.0.pdf

[9] M.-C. Chow, *Understanding SONET/SDH Standards and Applications*. Holmdel, NJ: Andan, 1995.

[10] I. Djordjevic, "LDPC coded modulation for terabit optical ethernet," *IEEE Photon. Soc. Summer Topicals*, Jul. 2011.

[11] D. Geisler, N. Fontaine, R. Scott, L. Paraschis, O. Gerstel, and S. Yoo, "Generation and detection of arbitrary modulation format, coherent optical waveforms scalable to a terahertz," *IEEE Photon. Soc. Summer Topicals*, Jul. 2011.

[12] S.-H. Fan, H.-C. Chien, J. Liu, and G.-K. Chang, "Toward terabit optical access network using spectrally-efficient, polarization-interleaved orthogonal wavelength-division-multiplexing," *IEEE Photon. Soc. Summer Topicals,* Jul. 2011.

[13] F. Kapron, D. Keck, and R. Maurer, "Radiation losses in glass optical waveguides," *Appl. Phys. Ltrs.*, vol. 17, no. 10, pp. 423-425, Nov. 1970.

[14] J. Buck, *Fundamentals of Optical Fibers*. Hoboken, NJ: John Wiley & Sons, 2004.

[15] G. Agrawal, *Fiber Optic Communication Systems*. New York: John Wiley & Sons, 1992.

[16] G. Goldfarb and G. Li, "Chromatic dispersion compensation using digital IIR filtering with coherent detection," *IEEE Photon. Technol. Ltrs*., vol. 19, no. 13, pp. 969-971, Jul. 2007.

[17] G. Agrawal, *Nonlinear Fiber Optics*. Burlington, MA: Academic Press, 2007.

[18] I.P. Kaminow and T.L. Koch, editors. *Optical Fiber Telecommunications IIIA*. Academic Press, San Diego, CA, 1997.

[19] E. Ip, J. Kahn, D. Anthon, and J. Hutchins, "Linewidth measurements of MEMS-based tunable lasers for phase-locking applications," *IEEE Photon. Ltrs.*, vol. 17, no. 10, pp. 2029-2031, Oct. 2005.

[20] E. Ip and J. Kahn, "Compensation of dispersion and nonlinear impairments using digital backpropagation," *J. Lightw. Technol.*, vol. 26, no. 20, pp. 3614-3425, Oct. 2008.

[21] R.-J. Essiambre, G. Kramer, P. Winzer, G. Foschini, and B. Goebel, "Capacity limits of optical fiber networks," *J. Lightw. Technol.*, vol. 28, no. 4, pp. 662-701, Feb. 2010.

[22] E. Ip and J. Kahn, "Digital equalization of chromatic dispersion and polarization mode dispersion," *J. Lightw. Technol.*, vol. 25, no. 8, pp. 2033-2043, Aug. 2007.

[23] S.L. Jansen, B. Spinnler, I. Morita, S. Randel, and H. Tanaka, "100GbE: QPSK versus OFDM," *Optical Fiber Technology, invited paper*, Jun. 2009.

[24] S.L. Jansen, I. Morita, T.C.W. Schenk, and H. Tanaka, "121.9 Gb/s PDM-OFDM transmission with 2-b/s/Hz spectral efficiency over 1000 km of SSMF," *J. Lightw. Technol.*, vol. 27, no. 3, pp. 177-188, Feb. 2009.

[25] B. Krongold and D. Jones, "PAR reduction in OFDM via active constellation extension," *IEEE Trans. Broadcast.*, vol. 49, no. 3, pp. 258-268, Sep. 2003.

[26] T. Jiang and Y. Wu, "An overview: peak-to-average power ratio reduction techniques for OFDM signals," *IEEE Trans. Broadcast.*, vol. 54, no. 2, pp. 257-268, Jun. 2008.

[27] R. van Nee, "OFDM codes for peak-to-average power reduction and error correction," *in Proc. IEEE Global Telecommun. Conf.*, 1996, pp. 740-744.

[28] L. N. Binh, T. L. Huyn, "Single and Dual-Level Minimum Shift Keying Optical Transmission Systems," *J. Lightw. Technol.*, vol. 27, no. 5, pp. 522-537, Mar. 2009.

[29] A. Stark, Y.-T. Hsueh, S. Searcy, C. Liu, A. McCurdy, R. Lingle, Jr., M. Filer, S. Tibuleac, G.-K. Cheng, and S. Ralph, "DQPSK for metro networks," *Proc. SPIE Photon. West*, 7621L, Jan. 2010.

[30] A. Stark, Y.-T. Hsueh, T. Detwiler, M. Filer, S. Tibuleac, A. McCurdy, R. Lingle, Jr., G.-K. Cheng, and S. Ralph, "Dispersion map optimization of single and dual-pol QPSK in the presence of aggressor channels," *IEEE Photon. Soc. Summer Topicals*, Jul. 2010.

[31] S. Searcy, A. Stark, Y.-T. Hsueh, T. Detwiler, S. Tibuleac, G.-K. Cheng, and S. Ralph, "Developing accurate simulations for high-speed fiber links," *Proc. SPIE Photon. West*, 79600A, Jan. 2011.

[32] A. Stark, Y.-T. Hsueh, S. Searcy, T. Detwiler, M. Filer, S. Tibuleac, G.-K. Cheng, and S. Ralph, "Scaling 112 Gb/s PDM-QPSK hybrid optical networks," *OFC 2011*, NWA4, Mar. 2011.

[33] Y.-T. Hsueh, A. Stark, M. Filer, T. Detwiler, S. Tibuleac, G.-K. Cheng, and S. Ralph, "Crosstalk-induced OSNR penalty prediction on 112 Gb/s PolMux-QPSK system," *CLEO 2011*, CThH6, May 2011.

[34] Y.-T. Hsueh, A. Stark, M. Filer, T. Detwiler, S. Tibuleac, G.-K. Cheng, and S. Ralph, "In-band crosstalk transmission penalties on 112 Gb/s PDM-QPSK optical links," *Photon. Technol. Ltrs.*, vol. 23, no. 11, pp. 745-747, Jun. 2011.

[35] A. Stark, Y.-T. Hsueh, T. Detwiler, M. Filer, S. Tibuleac, R. Lingle, Jr., G.-K. Cheng, and S. Ralph, "Filter bandwidth and subcarrier spacing tolerance of single-carrier 32 Gbaud PDM-QPSK," *IEEE Photon. Soc. Summer Topicals*, Jul. 2011.

[36] J. Anderson, T. Aulin, and C.-E. Sundberg, *Digital Phase Modulation*, New York: Plenum, 1986.

[37] T. Aulin, N. Rydbeck, and C.-E. Sundberg, "Continuous phase modulation – part II: partial response signaling," *IEEE Trans. Commun.*, vol. 29, no. 3, pp. 210-225, May 1981.

[38] P.A. Laurent, “Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (AMP)," *IEEE Trans. Commun*., vol. 34, no. 2, pp. 150-160, Feb. 1986.

[39] U. Mengali and M. Morelli, “Decomposition of M-ary CPM signals into PAM waveforms," *IEEE Trans. Inform. Theory*, vol. 41, no. 5, pp. 1265-1275, Sep. 1995.

[40] K.-P. Ho and J. Kahn, "Channel capacity of WDM systems using constant-intensity modulation formats," *OFC 2002*, pp. 731- 733, Mar. 2002.

[41] J. Tan and G. Stüber, "Frequency-domain equalization for continuous phase modulation," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2479-2490, Sep. 2005.

[42] W. van Thillo, F. Horlin, J. Nsenga, V. Ramon, A. Bourdoux, R. Lauwereins, "Low-Complexity Linear Frequency Domain Equalization for Continuous Phase Modulation," *IEEE Trans. Wireless Commun*., vol. 8, no. 3, pp. 1435-1445, Mar. 2009.

[43] X. Xu, H. Ma, H. Zhang, Z. Song, and Y. Cai, " Phase and Timing Recovery based on Frame Synchronization and Fractional-Spaced Frequency Domain Equalization for CPM," *in Proc. ISCIT 2009*, pp.1477-1482, Sep. 2009

[44] Q. Zhao and G. Stuber, "Robust time and phase synchronization for continuous phase modulation," *IEEE Trans. Commun.*, vol. 54, no. 10, pp. 1857-1869, Oct. 2006.

[45] M. Sandell, J. van de Beek, and P. Börjesson, "Timing and frequency synchronization in OFDM systems using the cyclic prefix," *in Proc. Int. Symp. on Synchronization*, Germany, 1995, pp. 16-19.

[46] Q. Zhao and G. Stuber, "Turbo synchronization for serially concatenated CPM," *in Proc. IEEE ICC 2006*, pp. 2976-2980, Jun. 2006.

[47] A. Graell i Amat, C. Nour, and C. Douillard, "Serially concatenated continuous phase modulation for satellite communications," *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 3260-3269, Jun. 2009.

[48] S. Cheng, M. Valenti, and D. Torrieri, "Coherent continuous-phase frequency-shift keying: parameter optimization and code design," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1792-1802, Apr. 2009.

[49] K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*, New York: John Wiley & Sons, 1999.

[50] A. Ambardar, *Analog and Digital Signal Processing*, Boston: PWS, 1995.

[51] S. Savory, G. Gavioli, R. Killey, and P. Bayvel, "Transmission of 42.8 Gbit/s polarization multiplexed NRZ-QPSK over 6400 km of standard fiber with no dispersion compensation," *Proc. OFC*, OTuA1, Mar. 2007.

[52] T. Detwiler, S. Searcy, A. Stark, S. Ralph, and B. Basch, "Asynchronously sampled blind source separation for coherent optical links," *Proc. SPIE Photon. West*, Jan. 2011.

[53] D. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Trans. Commun.* vol. COM-28, no. 11, 1867-1875, Nov. 1980.

[54] K. Kikuchi, "Polarization-demultiplexing algorithm in the digital coherent receiver," *Proc. IEEE LEOS Summer Topicals*, 2008.

[55] M. Oerder and H. Myer, "Digital filter and square timing recovery," *IEEE Trans. Commun.* vol. COM-36, no. 5, 605-612, May 1988.

[56] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*, New York: John Wiley & Sons, 1998.

[57] R. Schafer and L. Rabiner, "A digital signal processing approach to interpolation," *Proc. IEEE*, vol. 61, pp. 692-702, Jun. 1973.

[58] C. Farrow, "A continuously variable digital delay element," *Proc. IEEE Intern. Symp. Circuits Syst.*, pp. 2641-2645, Jun. 1988.

[59] X. Zhou, X. Chen, W. Zhou, Y. Fan, H. Zhu, and Z. Li, "All-digital timing recovery and adaptive equalization for 112 Gbit/s POLMUX-NRZ-DQPSK optical coherent receivers," *J. Optical Commun. and Networking*, vol. 2, no. 11, pp. 984-990, Nov. 2010.

[60] F. Gardner, "A BPQK/QPSK timing-error detector for sampled receivers," *IEEE Trans. Commun.*, vol COM-34, no. 5, pp. 423-429, May 1986.

[61] M. Simon, "Nonlinear analysis of an absolute value type of an early-late gate bit synchronizer," *IEEE Trans. Commun.*, vol 18, no. 5, pp. 589-596, Oct. 1970.

[62] K. Mueller and M. Muller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, vol. Com-24, pp. 516-531, May 1976.

[63] A. J. Viterbi and A. M. Viterbi, "Nonlinear estimation of PSK-modulated carrier phase with application to burst digital transmission," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 4, Jul. 1983.

[64] M. Taylor, “Phase estimation methods for optical coherent detection using digital signal processing,” *J. Lightw. Technol.*, vol. 27, no. 7, pp. 901-914, Apr. 2009.

[65] T. Detwiler, S. Searcy, S. Ralph, and B. Basch, "Continuous phase modulation for fiber-optic links," *J. Lightw. Technol.*, JLT-12747-2010: accepted.

[66] S. Searcy, A. Stark, Y.-T. Hsueh, T. Detwiler, S. Tibuleac, G.-K. Cheng, and S. Ralph, "Developing accurate simulations for high-speed fiber links," *Proc. SPIE Photon. West*, Jan. 2011.

[67] T. Detwiler, S. Searcy, B. Basch, and S. Ralph, "Continuous phase modulation as an alternative to QPSK for 100 Gb/s optical links," *NFOEC*, NTuB4, Mar. 2010.

[68] T. Detwiler, A. Stark, Y.-T. Hsueh, S. Searcy, R. Lingle, Jr., B. Basch, G.-K. Cheng, and S. Ralph, “Offset QPSK for 112 Gb/s coherent optical links,” *IEEE Photon. Soc. Summer Topicals*, Jul. 2010.

[69] T. Detwiler, A. Stark, Y.-T. Hsueh, S. Searcy, B. Basch, R. Lingle, Jr., G.-K. Cheng, and S. Ralph, “Offset QPSK receiver implementation in 112 Gb/s coherent optical networks,” *ECOC 2010*, Sep. 2010.

[70] IEEE Std. Local and Metropolitan Area Networks, Part 3, Amd. 4, *IEEE Standard 802.3ba*, 2010.

[71] T. Detwiler, A. Stark, B. Basch, and S. Ralph, “DQPSK for terabit Ethernet in the 1310 nm band,” *IEEE Photon. Soc. Summer Topicals*, Jul. 2010.

[72] T. Detwiler, S. Searcy, R. Lingle, Jr., B. Basch, and S. Ralph, “Quaternary modulation formats for 100-Gbps optical links,” *Proc. SPIE Photon. West*, Jan. 2010.

[73] T. Detwiler, S. Searcy, B. Basch, and S. Ralph, “Block linear coherent detection of optical continuous phase modulation,” *IEEE Photon. Soc. Summer Topicals*, Jul. 2010.

[74] T. Detwiler, S. Searcy, A. Stark, B. Basch, and S. Ralph, “Avoiding fiber nonlinearities by choice of modulation format,” *Proc. SPIE Photon. West*, Jan. 2011.

[75] A. Stark, Y.-T. Hsueh, S. Searcy, T. Detwiler, S. Tibuleac, M. Filer, G.-K. Chang, and S. Ralph, “Scaling 100G QPSK links for reliable network development,” *Proc. SPIE Photon. West*, Jan. 2011.

[76] T. Sakamoto, A. Chiba, and T. Kawanishi, “High-bit-rate optical QAM,” *OFC 2009*, OWG5, Mar. 2009.

[77] Y. Cai, J. Cai, C. Davidson, D. Foursa, A. Lucero, O. Sinkin, A. Pilipetskii, G. Mohs, and N.S. Bergano, “Achieveing high spectral efficiency in long-haul transmission with pre-filtering and multi-symbol detection”, *Commun. and Photon. Conf. and Exhibition*, pp. 349, 2010.

[78] W. Van Thillo, J. Nsenga, R. Lauwereins, V. Ramon, A. Bourdoux, and F. Horlin, “A new symbol block construction for CPM with frequency domain equalization," *Proc. IEEE ICC*, May 2008.

[79] H. Bülow, F. Buchali, and A. Klekamp, "Electronic dispersion compensation," *J. Lightw. Technol.*, vol. 26, no. 1, Jan. 2008.

[80] A. Klein, G. Kaleh, and P. Baier, “Zero forcing and minimum mean-square-error equalization for multiuser detection in code-division multiple-access channels," *IEEE Trans. Veh. Technol*., vol. 45, no. 2, pp. 276-287, May 1996.

[81] G. Cariolaro, "A system-theory approach to decompose CPM signals into PAM waveforms," *IEEE Trans. Commun*., vol.58, no.1, pp. 200-210, Jan. 2010.